NETROLOGIC

INCORPORATED

PHASE I FINAL REPORT

Microcomputer-based Vehicle Routing and Scheduling

AFOSR SBIR 90-190
Contract Nr. F49620-90-C-0049

17 Jun 1991

To: USAF, AFSC
Air Force Office of Scientific Research
Building 410
Bolling AFB, DC 20332-6448

From: NETROLOGIC, INC.
5200 Springfield, Suite 312
Dayton, OH 45431

NETROLOGIC, INC.
5080 Shoreham Place, Suite 201
San Diego, CA 92122

Subcontractor: North Dakota State University
Dept of Computer Science and Operations Research
Fargo, ND 58105

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED
FINAL 01 July 90 to 31 Dec 90

4. TITLE AND SUBTITLE
MICROCOMPUTE-_BASED VECHILE ROUTING AND SCHEDULING

5. FUNDING NUMBERS
AFOSR-90-0190
F49620-90-C-0049
65502F 3005/A1

6. AUTHOR(S)
DR. JOHNSON

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NETROLOGIC, INC
5200 SPRINGFIELD, SUITE 132
DAYTON, OH 45431

8. PERFORMING ORGANIZATION REPORT NUMBER
AEOSR-TR- 91 0665

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
F49620-90-C-0049

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release;
distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Netrologic has designed and implemented a system that uses alternative ways of employing methods of artificial intelligence in conjunction with heuristic mathematical models for solving vechile routing problems and applied them to the Air Force LOGAIR cargo handling systems. The artificial intelligence problem solving techniques involve genetic algorithms and set partitioning algorithms as applied to the LOGAIR vehicle routing problem.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCALSSIFIED	UNCLASSIFIED	UL

Acknowledgements

Netrologic wishes to acknowledge the significant effort that went into this Phase I project by a number of researchers. Their efforts have contributed substantially to furtherment of an understanding of practical routing a scheduling solutions in more nearly real time. The following individuals contributed to the work under this program.

Dr. Kendall Nygard
North Dakota State University

Sam Thangiah
NDSU

Doug Shesvold
NDSU

Taesik Kim
NDSU

James R. Johnson
Netrologic

Tom Sears
Netrologic

Dr. Dapeng Li
Netrologic

We are also very grateful for the support from the Air Force Office of Scientific Resesearch without whose generous support this work would not have been possible.

Table of Contents

Microcomputer-based Vehicle Routing and Scheduling

List of Figures

List of Tables

Abstract

During Phase I Netrologic and Dr. Nygard's team at NDSU developed routing and scheduling algorithms which advanced the state of the art in efficient heuristic route development for computationally NP hard problems. Four seperate genetic-algorithm route development techniques were designed with very fast evaluation functions. These route development techniques used the multiple evaluation functions to perform parallel searches which competed for the best solution. Current best solutions found by one method were shared with all methods to enanble them to explore promising new areas in the search hyperspace. After convergence on a "best" solution a genetic algorithm post processor, XCHANGE, swapped stops among adjacent routes to obtain further route efficiencies by exploring local optimizations. The result of the multistage search was an efficient search which in every case tested achieved better results than the best known technique previously tried. Typical improvements in efficiency over other techniques was about 3 percent.

When these techniques were applied to the Air Force LOGAIR problem an iterative approach was used. First, good solutions were found for trunk routes between the six major depots. Next clusters of bases were assigned to each depot and then a Traveling Salesman Problem was developed which assigned stops in a depot's cluster to a specific sequence for that route. The problem was iterated by going back and adjusting the route structure between depots and then repeating the depot clustering and stop scheduling process.

These techniques augment a cargo allocation system which was previously implemented to maximize cargo loaded onto aircraft at a depot. With the cargo loading expert system and the scheduling assistant LOGAIR could theoretically achieve a 13 percent improvement in cargo handling at no increase in cost. This translates into millions of dollars saved on an annual basis.

A graphical user interface prototype model was developed for LOGAIR use. It showed route structures and enabled users to manually edit routes and quickly observe results. Animation enabled the user to see proposed cargo allocation solutions in action.

In Phase II we will extend the vehicle routing capability developed under Phase I to the multi-depot problem. We will address routing problems with time constraints, and we will develop a dynamic rescheduling capability.

FINAL REPORT

Microcomputer-based Vehicle Routing and Scheduling
SBIR AF90-190

1.0 Introduction

We have designed and implemented a system that uses alternative ways of employing methods of artificial intelligence in conjunction with heuristic mathematical models for solving vehicle routing problems and applied them to the Air Force LOGAIR cargo handling system. The artificial intelligence problem-solving techniques involve genetic algorithms and set partitioning algorithms as applied to the LOGAIR vehicle routing problem.

It is well known that many problems in Operations Research are NP complete and therefore cannot be solved to optimality in realistic computer time. For such problems, researchers have developed many heuristic strategies. A major problem with many of these heuristics is an inability to deduce the kind of strategy to adopt, given the characteristics of the problem at hand. The need to deduce the parameters to use in a heuristic for a given problem led to the identification and use of certain AI problem solving techniques in order to supply answers to what values of parameters were required.

1.1 The Problem Solving Techniques

We have applied the following problem solving techniques:

- **Genetic algorithms** are search techniques which belong to the "generate and test" AI paradigm. Genetic algorithms (GA) (Holland, 1975) base their search for better solutions on principles of survival of the fittest genes, where the genes represent possible solutions to the problem and the fitness is the designer-specified objective function value.

- **Heuristic Mathematical models** are instruments which transform data into information which can aid in the inferencing mechanism. In addition, many real world situations can be modeled with sets of equations. Thus, good mathematical models are fundamentally important pieces of knowledge in the system.

1.2 The Problem Domain: The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a highly combinatorial problem (NP-Hard) that has been extensively studied by Operations Researchers (Bodin *et al.*, 1983). In the VRP there is a known collection of stop points that have demands for service, and a fixed fleet of limited capacity vehicles to serve the stops. The problem is to find the minimum-distance way to assign the stops to vehicles and specify the orders in which each of the vehicles visits its stop. All the vehicles begin and end their tours at a fixed location depot.

In essence, the AI tools operate as an intelligent controller, first devising an advanced set of candidate solutions, then adaptively guiding the solution procedure used by a mathematical algorithm. The resultant solution methods dynamically specify local details and are adaptive to a wide variety of problem instances. In many cases, the methods discussed produce superior solutions to routing and scheduling problems in relatively modest CPU time.

Genetic Algorithms are generally compute-intensive procedures that require the evaluation of many candidate solutions to a given problem. To reduce The computational overhead of this approach, a mechanism for improving the performance of the genetic search has been developed that uses multiple evaluation functions, permitting the parallel investigation of multiple peaks in the search space.

The purpose of the XVRP-GA system is to assist researchers and decision makers in applying mathematical models to a specific problem instance by "tuning" the mathematical models to the problem description, and adaptively "steering" the mathematical model as a solution evolves. The structural system overview of XVRPGA is discussed in section 2. Sections 3 and 4 provide the genetic algorithm based system module. Section 5 describes the results and comparisons of the XVRP-GA system with alternative methods in computer aided vehicle routing.

2.0 Genetic Algorithm Methods Application to the Air Force LOGAIR System

In our development of routing and scheduling algorithms under this program we have been interested in more than a theoretical development. We have continually kept in mind the practical applications of the technology and, indeed, developed a practical prototype system for use with the Air Force's LOGAIR system. During Phase II this will be developed into a complete workable system for operational use with the LOGAIR system. The theoretical and practical achievements of Phase I routing and schdeuling effort are described in the following sections.

2.1 Problem Description

LOGAIR is a domestic airline system that facilitates the movement of cargo between Air Force bases and depots in the United States. There are 6 of these depots or Air Logistics Centers (ALCs) and 46 other bases in the U.S. which constitute the LOGAIR system. The 6 ALCs are major repair and supply facilities that service the other Air Force bases. These facilities are responsible for the supply and maintenance of serviceable spares for all aircraft, missiles and ground radar systems. This results in the need to annually ship thousands of tons of cargo between the various ALCs and bases. In 19XX, over 132,000 tons of cargo was moved between pairs of bases. The Air Force contracts with commercial cargo airline companies to fly fixed routes among the ALCs and Air Force bases.

From year to year the shipping requirements between the bases may change. Consequently, each fiscal year, the routes flown in the LOGAIR system are modified to reflect these forecasted changes. Route changes are not permitted during the fiscal year due to the contracting process. Cargo airline companies base their bids on the routes specified by LOGAIR personnel. Any modifications made during the year would require renegotiation of contracts as well as budget changes resulting from this renegotiation.

There are several factors that are considered in the annual route design process. The main objective is to move as much of the highest priority cargo as possible while keeping costs at a minimum. The goal for cargo movement is within 36 hours. The various costs associated with the shipping of cargo include mileage rates and fuel consumption for the different types of aircraft, transportation taxes and landing fees. Table 1 shows the mileage rates and the fuel consumption of the available types of aircraft. The transportation tax is 6.25% of the mileage cost per year, excluding fuel. The L100 is the only aircraft has a landing fee, which is $250 per landing. There are certain restrictions on the routes that may be flown. The amount of cargo on an aircraft at any point in time cannot exceed the capacity of the aircraft. Also, there are limits on the amount of time a crew can fly and work in a single day. Consequently, the speed of the available types of aircraft also becomes an important issue as longer routes would need faster aircraft.

AIRCRAFT CHARACTERISTICS		
AIRCRAFT TYPE	MILEAGE RATE	FUEL (gals / mile)
L100	$ 7.6940	2.6000
L188	$ 6.4344	2.5910
DC-9	$ 5.7449	2.1470
CV 640	$ 5.9741	1.8750

Table 1. Aircraft Characteristics For Determining Aircraft Assignments To Routes

2.1.1 Present LOGAIR Operation

Currently, there are two distinct areas in the decision making processes of the LOGAIR operation. The first is the aforementioned annual route design. The second is the daily allocation of cargo to be moved given the fixed routes. At present, LOGAIR personnel have computerized decision support systems in both of these areas.

2.1.1.1 Annual Route Design

The decision support system for the route design process has two separate components. The first component is the route generator. It relies on a continuous multicommodity network formulation of the problem to generate candidate routes. This model requires as input, a forecast matrix and a distance matrix. The forecast matrix gives the total pounds of cargo which must be moved from each base to each other base each day. The distance matrix gives the flight distance between all pairs of bases. The route generator produces a nominal set of cargo routes which minimizes pound-miles. This set of routes may violate various constraints. LOGAIR personnel then modify these routes as needed to form a set of feasible routes. This set of potential routes is then the input for the second component. The second component is the route selector. It relies on a fixed charge multicommodity network formulation to select a subset of the potential routes. This subset of the input feasible routes is the optimum set of routes which will guarantee that the daily demand is met subject to aircraft capacity constraints.

2.1.1.2 The LOGAIR ALLOCATE System

The LOGAIR flight system moves cargo among 56 Air Force Bases in the continental United States each day. There are 16 flights, and each day there is always considerably more air-eligible cargo than can be transported by LOGAIR. This section describes ALLOCATE, a comprehensive computer-based system for allocating cargo pallets to aircraft.

ALLOCATE has several specialized reports and features designed to closely involve the user with the process of developing an allocation. However, the heart of the system is an automatic allocator which combines a generalized assignment model named ASSIGN and an expert system named REVISE. ASSIGN handles the allocation fundamentals, including ensuring that pallets are allocated to only one flight, reach the proper destinations, and do not exceed aircraft capabilities. REVISE handles a host of complications, including restrictions on the ways that pallets can be arranged on aircraft types, and limitations on transporting hazardous cargo.

2.1.1.2.1 The LOGAIR Problem

The LOGAIR system is illustrated in Figure 1. The 9 large nodes represent major maintenance and repair facilities called Air Logistic Centers (ALCs) and points of embarkation for overseas flights. Interconnecting these large stations are 6 trunk routes providing numerous pallet movement options, The other 10 flights are petal routes, loops which originate and terminate at an ALC. The routes flown on a given day are fixed, but four different route systems are used each week. The routes are redesigned once each year.

In advance of the flight departures on a given day, each station uses a computer communications network to send a message called a Cargo Requirements Report (CRR) to Wright-Patterson Air Force Base in Ohio. A CRR indicates the number and types of pallets which the station anticipates having ready to send to various other stations that day. LOGAIR controllers at Wright-Patterson allocate the pallets to flights and send the allocations (called load reports) over the network to the stations.

The cargo allocation process is characterized by the following factors:

1. Pallets are categorized into high, medium and low priority classes.

2. Often there are several alternative flights to which particular pallets can be assigned.
3. Pallets are of 2 basic sizes, with one large pallet being equal in length to a small pallet but having twice the width.

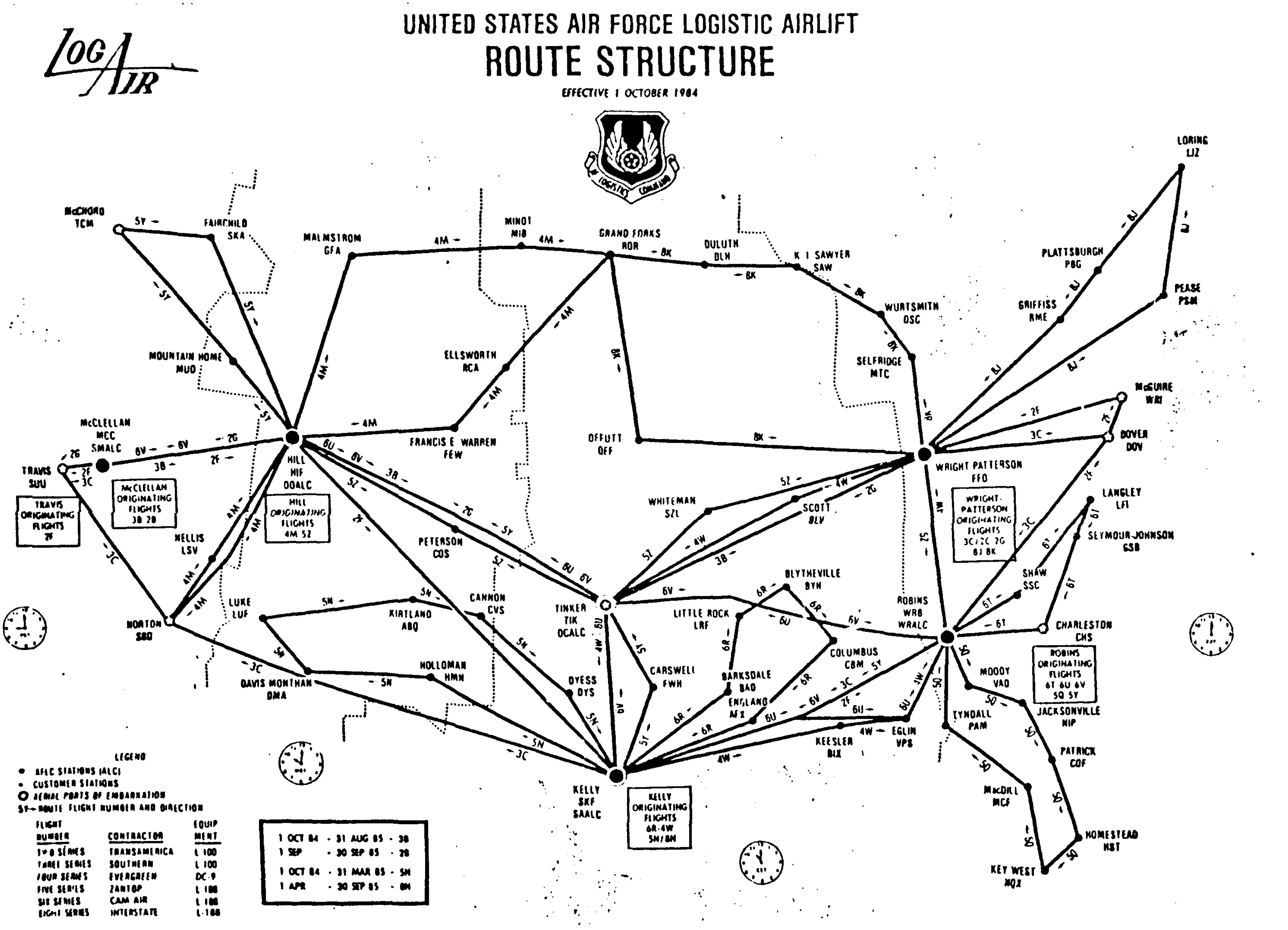
4. Pallets of the same size are often clamped together to form in-line pallets used to transport long items.

5. Three aircraft types are currently in use (Lockheed L-100 and L-188 and McDonnellDouglas DC-9), and capacities differ among the types.

6. There are always many more pallets available than can possibly be allocated. As a result, low priority pallets are often diverted to surface transportation, and some pallets must wait and be allocated on a later data.

7. Each aircraft type has geometric layout and loading limitations. For example, the DC-9 aircraft (illustrated in Figure 2), has side loading doors which cannot accommodate in-line pallets longer than 3 units (large or small), and a tapered cargo compartment which allows only small pallets at the ends. In additional many combinations of pallet sizes (especially in-line pallets), cannot be accommodated.

8. Pallets may hold hazardous cargo which is categorized into 29 possible types. Incompatible types cannot be allocated to the same aircraft, some types are allowed to land only at authorized stations, and there are weight limits by hazardous type which apply to both aircraft and to certain stations.



FLIGHT NUMBER	CONTRACTOR	EQUIPMENT
1 & 0 SERIES	TRANSAMERICA	L 100
THREE SERIES	SOUTHERN	L 100
FOUR SERIES	EVERGREEN	DC-9
FIVE SERIES	ZANTOP	L 188
SIX SERIES	CAM AIR	L 188
EIGHT SERIES	INTERSTATE	L-188

Figure 1. United States Air Force Logistic Airlift Route Structure .

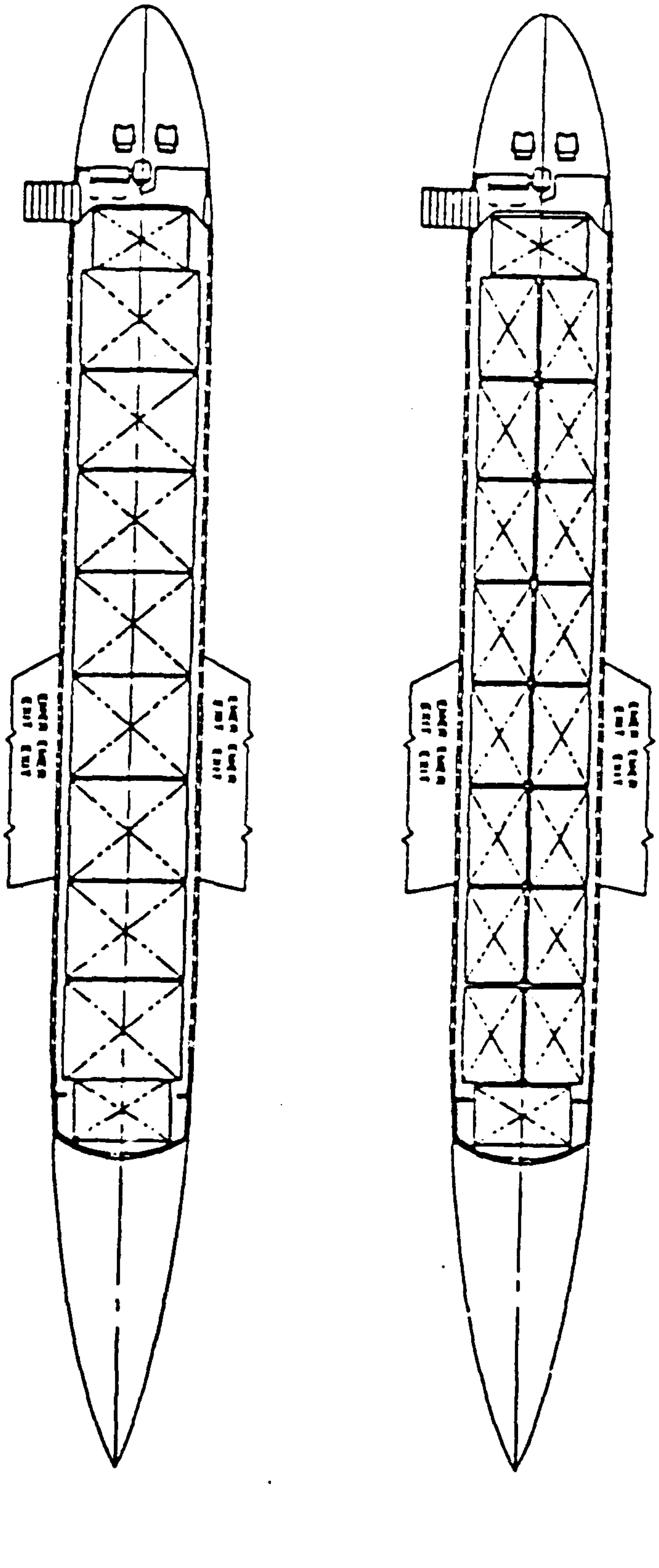


Figure 2. Loading Arrangements in the DC-9 Aircraft

9. There are circumstances under which cargo is allowed to be shipped to an intermediate station rather than a final station.

The goal is to maximize the number of high priority pallets allocated, followed in turn by the medium and finally the low priority pallets. If there are alternative ways to allocate the same number of pallets in a priority class, a secondary criteria of minimizing total pallet-miles is appropriate.

This goal together with factors 1-6 led to the ASSIGN model, which captures much of what constitutes a good allocation. The REVISE expert system modifies the allocation produced by ASSIGN and renders the allocation feasible with respect to factors 7-9.

2.1.1.2.2 Overview of ALLOCATE

ALLOCATE is a comprehensive decision support system for the cargo allocation process. Figure 3 below illustrates that ALLOCATE has a top level Command Processor from which the user issues commands to carry out various Support Functions or invokes the automatic Pallet Allocator.

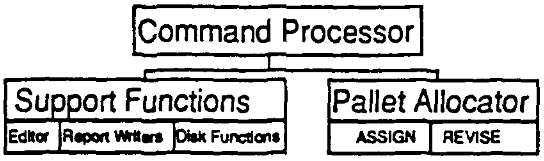


Figure 3. High Level Functional Diagram Of ALLOCATE Cargo Handling System

The Support Functions are tools which facilitate human control of the allocation process. The primary Support Functions and brief description of typical usage follow:

1. An Editor, from which the user can allocate pallets manually change characteristics of pallets and flights, and enter new pallets into the ALLOCATE system. A typical user, at least when first learning the system, may elect to allocate some or all pallets directly from the editor essentially using ALLOCATE as an automated version of the old system.

2. Report writers, including both printed and screen versions of official forms (the load report and a report of unallocated pallets), and a worksheet form which was used for allocating in the old system. All the forms are available upon command at any point in the allocation process, making it easy for the user to monitor progress.

3. Disk Functions. which allow the user to save and retrieve allocations to and from disk at any point in the process. Any number of allocations can be saved by name, allowing the user to develop and retain a collection of tentative allocations before committing to a final decision.

The Pallet Allocator uses a Generalized assignment mathematical model called ASSIGN and an expert system called REVISE to automatically produce an allocation of pallets to aircraft. The work is carried out in 2 major steps. First, ASSIGN produces an allocation which considers factors 1-6 above, but ignores factors 5, 6 and 7. ASSIGN uses a sophisticated generalized assignment model which maximizes the number of pallets allocated as a primary criteria and minimizes total pallet miles as a secondary criteria. The solution algorithm is an advanced branch and bound method which uses a specialized Lagrangian relaxation with multiplier adjustment. Second, the expert system REVISE begins with the allocation produced by ASSIGN, and applies production sysiem rules involving factors 7-9 to generate a final allocation. Details of ASSIGN and REVISE are provided in the next 2 sections.

The user has options of requesting: i) an optimal allocation or ii) a quick allocation. Both options also allow allocation of all priority classes in turn, or sequential allocation of all priority classes. The quick allocation is provided much faster by a heuristic solution of the underlying generalized assignment problem.

After the allocation is produced, control returns to the Command Processor. The user can elect to use Support Functions to evaluate, store, revise, or finalize an allocation at any time. For example, under unusual circumstances (e.g., closed runways adverse weather, disabled aircraft, emergency shipments. etc.) the user may wish to use the editor to manually allocate or deallocate pallets either before or after running the automatic Pallet Allocator. Allocation decisions are ultimately finalized by, and are the responsibility of, a human, but are substantially aided by a sophisticated model, expert system, and the Support Functions.

2.2 Alternative Systems for Route Design

To assist the LOGAIR planners in their annual route replanning task Netrologic and North Dakota State University (NDSU) designed two different systems for optimizing route design that employ alternative search strategies. These will be described in detail along with a description of the supporting graphics user interface environment.

2.2.1 Genetic Search Approach

The first alternative system to be explored is based on genetic search techniques. The basic idea of the genetic search technique is to mimic the process of reproduction of fittest members as is observed in real biological systems in hopes of generating high performance solutions for a particular problem. The second alternative system to be explored is based on a set partitioning formulation. Like the current manual system, our techniques generate a set of feasible routes from which a subset is chosen to provide a high performance solution.

2.2.1.1 The Genetic Search Idea

The genetic search technique involves simulating a biological process of reproductive permutation of the gene pool and survival of the fittest where route solutions are represented as a genetic pattern. This technique is an iterative procedure which maintains a population of solutions to the objective function of interest. Each member of the population is a binary string of some fixed length. This implies that there is some function which maps from the set of all binary strings of this fixed length to the set of all solutions in the search space. This function must be chosen very carefully so that substrings in the binary string represent substructures that carry pertinent information about the solution. These substructures are analogous to the genetic material found in biology. At each iteration or generation some members of the population are selected to reproduce. Preference is given to members which exhibit high performance as measured by the objective function. In keeping with the technology of genetics this is often referred to as the "fitness" function. This reproduction is accomplished by cutting the binary strings at certain points, exchanging the resulting substrings with another member of the population, and recombining to form new population members. The idea in genetic search is that after simulating many generations the superior substructures will occur with great frequency in the members of the population because of their contribution to the overall performance of the solutions. Eventually various superior substructures will combine to form high performance solutions.

Genetic algorithms (GA) are heuristic solvers of combinatorial problems that proceed in a manner inspired by biological genetics (Goldberg, 1989). Basically, candidate solutions to the problem are represented as bit strings (chromosomes), and populations of these solutions are simulated over some number of generations, seeking a high quality solution to the problem of interest. Criteria that involve "survival of the fittest" concepts provide the pressure for populations to develop increasingly fit individuals. The GA exploits the accumulating knowledge of the Vehicle Routing Problem (VRP) solution being explored. Each point in the control parameter space is a genetic string represented as a binary number. Each string has a field allocated for the performance function, F_e, which is returned by the evaluation function. For each generation the GA maintains a population of these control parameter strings. Each individual population member is evaluated as a set of control parameters and the associated performance measure is saved. Finally, using selection probabilities these control parameter strings undergo reproduction via the crossover and mutation genetic operators. Although there are many variants, the basic structure of the genetic algorithm is shown in Figure 4.

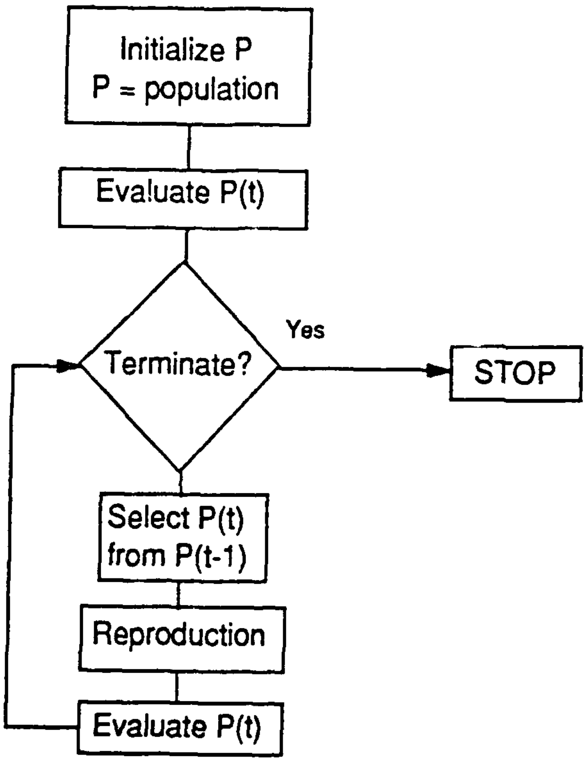


Figure 4: The basic structure of the genetic algorithm. Genetic algorithms are heuristic solvers of combinatorial problems that proceed in a manner inspired by biological genetics.

2.2.1.2 Genetic Algorithm Operators

Genetic Algorithms are generally compute-intensive procedures that require the evaluation of many candidate solutions to a given problem. In the past few years many researchers have investigated ways of improving the performance of GAs through the development of more efficient genetic alteration techniques. Since the primary parameters of a standard GA are population size, crossover, mutation rates, and number of crossover points, significant attention has been paid to these parameters to improve performance and efficiency. New techniques to improve selection of these parameters have had a considerable impact on performance (Schaffer, 1989; Goldberg, 1989; Jog, 1989). Adaptive selection methods (Baker, 1985) and reproductive evaluation techniques (Whitley, 1987) have also been shown to speed up GA searches. In the application area of routing and scheduling, genetic algorithms set parameters for a mathematical heuristic. To reduce the computational overhead of this approach, a mechanism for improving the performance of the genetic search is detailed in this section. A method of using multiple sharing evaluation functions is employed, permitting the parallel investigation of multiple peaks in the search space.

2.2.2 System Overview of XVRP

The XVRP system improves the ability of heuristic procedures to design routing plans by intelligently setting their parameters. The overall system starts with problem data consisting of stop-point locations, depot location, number of vehicles and their capacities. The structural description of each of the basic functional components is described in this section. Figure 5 provides a detailed view of the basic components of XVRP system.

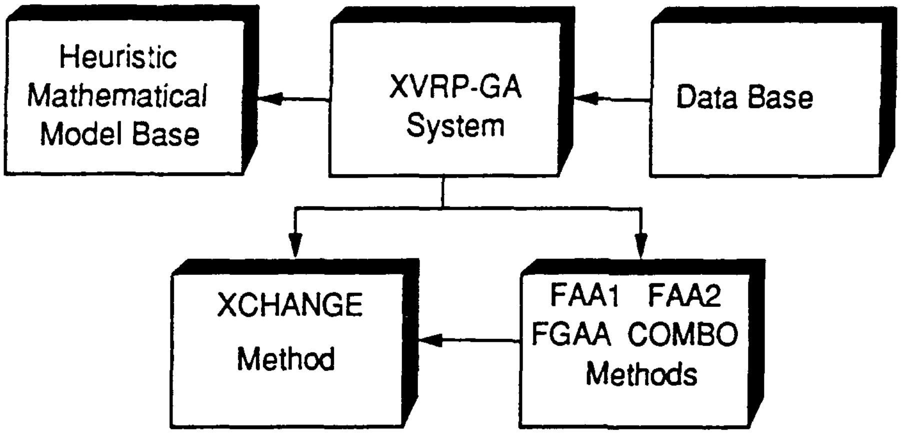

Heuristic Mathematical Model Base
XVRP-GA System
Data Base
XCHANGE Method
FAA1 FAA2 FGAA COMBO Methods

Figure 5: The overall XVRP system. The system improves the ability of heuristic procedures to design routing plans by intelligently setting their parameters.

We use a "cluster first / route second" heuristic technique as a VRP solver. There are two fundamental decisions that must be made before using this heuristic. First, we need to determine the clusters by some method. Second, we need to sequence the stop locations in a cost effective way. The objective is to find the order in which the stops are visited so that the total distance traveled is as small as possible. Four methods of determining the clusters have been developed. The first two methods, Fast Assignment Approaches FAA1 and FAA2, are new algorithms that are relatively fast and well suited for the genetic search. The third method, FGAA, is a modified version of the generalized assignment method developed by Fisher and Jaikumar (1981). The fourth method, COMBO, is a combination method, which uses the same GA recommended seed points for the three methods mentioned above (FAA1, FAA2 and FGAA), but selects best solutions from the currently best performing method during the route development process. A local optimization process is then used on the "good" clusters recommended by the COMBO method to explore the possibility of better solutions in the vicinity of the "good" solution.

After the GA methods have selected a good candidate route structure a postprocessor is run on the candidate solution. The postporcessor, XCHANGE, is a simple postprocessor, but achieves excellent power from the use of genetic algorithms. The XCHANGE method uses the genetic string to interpret the stop assignments of a particular route. The effect of the genetic recombinations is to make simple alterations to the existing system of routes.

2.2.3 A Genetic Search Strategy for LOGAIR

In the LOGAIR problem, there are 6 ALCs and 48 other airforce bases. Any routes that are flown between the ALCs are referred to as "trunk" routes and the the remaining routes are referred to as "feeder" routes. A solution to the routing problem will be obtained in two phases. In the first phase, the feeder routes are fixed while a genetic search is conducted to find the best set of trunk routes for the given feeder routes. Once this phase is complete, the trunk routes obtained in the first phase are fixed while a genetic search is conducted to obtain the best set of feeder routes for the given trunk routes. A population member or routing solution is evaluated according to the total amount of cargo moved. Thus the algorithm proceeds to search for a routing solution that maximizes the amount of cargo moved. The input to the algorithm includes a forecast matrix, a distance matrix, characteristics of available aircraft for trunk and feeder routes, and an initial set of feeder routes.

2.2.3.1 The Trunk Routing Procedure

The problem addressed in the trunk routing procedure is to find a set of routes to fly between the ALCs. The six ALCs are MCC, WRB, FFO, TIK, HIF and SKF (McClellan AFB, Warner Robins AFB, Wright Patterson AFB, Tinker AFB, Hill AFB and Kelly AFB, respectively). For the purposes of the genetic search, the number of aircraft which are used for the trunk routes is assumed to be fixed at some number N. Each of the N routes is represented in the chromosome by an 18 bit string. Thus, each chromosome is of length 18 * N and represents an entire trunk routing plan. In the chromosome representation the integer values that are used to represent the ALCs are 1 = MCC, 2 = WRB, 3 = FFO, 4 = TIK, 5 = HIF, and 6 = SKF. Each consecutive 3 bit string is interpreted as the binary representation of an integer in the range of 0-7. Thus each 3 bit string represents an ALC except for the the strings "000" = O and " 111 " = 7 which do not correspond to any of the ALCs. These strings are interpreted as null bases. Each 18 bit string (3 bits per base times 6 bases) is interpreted as a sequence of ALCs which form a trunk route. As an example, suppose the integer sequence derived from an 18 bit string is 2-1-4-5-6-7. The route flown by this aircraft would be WRB-MCC-TIK-HIF-SKF. Notice that there are only 5 ALCs visited by this aircraft. This is because the integer value of 7 is interpreted as a null base. If there are a number of consecutive 3 bit strings that are identical, they are interpreted as just one 3 bit string. For example, suppose the integer sequence derived from an 18 bit string is 4-4-1-4-5-7. This string would be interpreted as 4-1-4-5-7, which translates into the route TIK-MCC-TIK-HIF.

Each population member determines a sequence of trunk routes, which together with the given Feeder routes form a network for a Multi-commodity Capacitated Transshipment Problem (MCTP). The information on available aircraft is read in from the feeder aircraft file and the trunk aircraft file. The ith trunk route in the sequence is matched with the ith aircraft listed in the trunk aircraft file. The aircraft types assigned to the routes determine the capacity of the arcs. Each of the 54 bases is associated with a different commodity, which represents cargo that originates from that particular base. The forecast matrix represents the demand of each base for cargo originating at each other base. This MCTP is solved with the objective being to maximize the amount of cargo moved. The amount of cargo moved is then the fitness value for a particular population member. Each population member is evaluated in this way for every generation.

2.2.3.2 The Feeder Routing Procedure

As was mentioned above, in the feeder route phase, the trunk routes are fixed while a genetic search is made for the feeder routes. The chromosomes or population members in this phase will represent various clusterings of the feeder bases to form feeder routes. This representation is described in [Thangiah *et al.*, 1990]. Each chromosome divides the feeder bases into a sequence of clusters, using a process called genetic sectoring. The number of clusters is equal to the number of aircraft that are available in the feeder aircraft route. This file contains a list of aircraft, each of which has attributes of capacity and its home base or ALC. Once the sequence of clusters has been calculated for a particular chromosome, each of the clusters is matched with an aircraft from the feeder aircraft file. The ith cluster is matched with the ith aircraft listed in the file. In this way, each cluster of feeder bases is assigned a depot or ALC. For each cluster and associated ALC, a selection/insertion algorithm is used to construct a tour or feeder route. The cheapest insertion rule is used with the objective being to minimize route distance. Once the feeder routes have been calculated for a particular population member, the resulting MCTP can be solved as in the trunk phase to obtain a fitness value. Again, this fitness value is a measure of how effective the route structure is relative to the goal of maximizing cargo movement.

As in all genetic algorithms, a key issue is the representation of information and how it is embedded in the chromosomes. In the feeder route phase of the genetic search, information describing a sequence of clusters is embedded in each chromosome. We now describe the way in which this information is represented. The idea for dividing the feeder routes into clusters is to establish a polar coordinate system for the feeder bases and partition the set of feeder bases using a set of angles. In this application, the pole or origin is taken to be Tinker Air Force Base and the zero angle is defined by a ray that originates from the origin and points due east. Angles increase in the counter-clockwise direction. The list of feeder bases is then ordered in terms of increasing polar angle. If there are K aircraft in the feeder aircraft file, then the Length of each chromosome will be (K-1) * 3 bits. Each consecutive 3 bit string represents an angle. Thus each chromosome can be decoded into a set of K-1 angles, which together with the zero angle partition the feeder bases into K sectors, one for each aircraft. The ith 3 bit sequence in the chromosome, given by B(i), is converted to an angle S(i+1) using the following formula:

$$S(i+1) = (i * \mathrm{MaxAngle} / K) + \mathrm{INT}(B(i)) * C.$$

Maxangle is the maximum polar angle among the feeder bases. INT is a function that converts a binary string into an integer value. The initial seed angle S(1) is assumed to be 0. As 3 bits give an integer value between 0 and 7, the value C is used to provide an increase in its range. If the second term of the sum were ignored, the feeder bases would be partitioned into sectors of equal size. The second term allows the boundaries between the sectors to deviate from this equal partition. It is these deviations that are encoded into the chromosomes. Once the sectors are formed for a given chromosome, the algorithm assigns every feeder base, f(i), to a cluster A(j), using the following criterion:

$$f(i) \text{ is assigned to } A(j) \text{ if } S(j) < \mathrm{PolarAngle}(f(i)) <= S(j+1),$$

where i = 1 to N (N = the number of feeder bases),

j = 1 to K - 1 and

PolarAngle is a function that returns the polar angle of the given feeder base.

As the reader can see from the description above, each chromosome determines a clustering of bases into feeder routes which is evaluated by its corresponding fitness function. After many generations, feeder routes emerge which exhibit high performance relative to the goal of maximizing cargo movement.

2.2.4 Functional Description of the Multiple Evaluation Functions for the GA

One of the major problems with genetic algorithms is to prevent the search from converging prematurely on less than global optimal solutions. Our contribution to solving this problem is to maintain competing solutions distributed through various "good" hyperplanes which maintain parallel competing processes throughout the genetic search. New "best" solutions are shared with all competing processes to enable them to find new good places to look. This section will detail the process of using multiple evaluation functions and the problem of representation.

2.2.4.1 Representation

The representation of a problem as an artificial chromosome is the key that characterizes an optimization problem as "GA easy" or "GA hard" (Liepins, GE., Vose M.D., 1990). In the parameter discovery task we seek a set of parameters under the guidance of a evaluation function. GAs use bit strings as chromosomal encodings of parameters of the problem they are trying to solve. The control parameters in our case are seed-point locations (Nygard, Juell and Kadaba 1989). We model each vehicle tour with a location called a seed, with the vehicle conceptually traveling from the depot to the seed and back. The seed-points represent an average location around which the aircraft serves. We use binary bit strings to encode the seed-points as shown in Figure 3. The number of seed-points recommended by the GA is equal to the number of vehicles available. The example string shown below represents three seed-points, each with an x and y coordinate, shown in both binary and decimal. The control parameters $S_{x1}, S_{y1}, S_{x2}, S_{y2},..S_{xN}, S_{yN}$ represent the N seed-points. The parameters are constrained to an interval of the form $a_i < S_N < b_i$ where a_i is 0 and b_i is 1023. Each seed-point is identified by an x-coordinate and a y-coordinate which consist of a 10 bit binary string to represent the numbers 0 to 1023 inclusive. The delivery locations are scaled to a 1024 X 1024 grid.

The length of the GA string is $L_{GA} = \sum_{j=1}^{N} L_j$

where L_j is the length of one parameter string.

An example string and the decoded parameters are shown below:

String:	0000101100	1010110010	1000110110	110010010	0111110010	00101101100
Parameter:	S_{x1}	S_{y1}	S_{x2}	S_{y2}	S_{x3}	S_{y3}
Decoded Value:	55	803	987	739	348	200

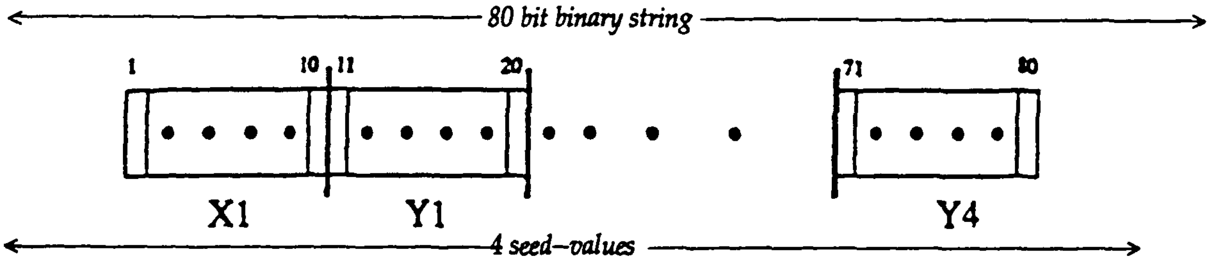


Figure 6: Each seed-point is identified by an x-coordinate and a y-coordinate which is represented by a 10 bit binary string to represent the numbers 0 to 1023 inclusive.

2.2.4.2 The Evaluation Functions

Several "cluster first, route second" heuristic techniques are available for use as VRP solvers, serving as fitness evaluation functions for the GA. There are two fundamental decisions that must be made by these heuristics. First, they must determine the clusters by some method. Second, they need to sequence the stop locations in a cost effective way. To utilize these hueristics a Traveling Salesman Problem (TSP) is constructed and used for sequencing the stops. In the TSP, a tour begins at a home location, visits each stop on a list exactly once, then returns to the location of origin. The objective is to find the order in which the stops are visited so that the total distance traveled is as small as possible. In the Euclidean TSP, each stop is identified by a coordinate location, and distances between stops are calculated by the Euclidean (or as the crow flies) metric. Tour construction algorithms are a prominent and successful class of heuristic procedures for quickly solving these types of large-scale instances of the TSP (Golden and Stewart, 1985). These methods construct tours incrementally starting with an initial subtour which is then expanded by repeatedly applying rules that select unvisited stops and which insert them into the tour until a solution is formed that visits all stops. The steps shown in Figure 7 indicate the method of using the GA to set the parameters for the heuristic mathematical models in the XVRP-GA module. We experimented with four methods of determining the clusters. The first two methods, Fast

Assignment Approaches FAA1 and FAA2, are new algorithms that are relatively fast and well suited for the genetic search. The third method, FGAA, is a modified version of the generalized assignment method developed by Fisher and Jaikumar (1981). The fourth method is a combination method, which uses the same GA recommended seed-points for the three methods mentioned above. Each of the methods is described below.

Stage 1: Accept the seed-points recommended by the Genetic Algorithm.
Stage 2: Use FAA1, FAA2 or the FGAA method shown below to determine the clusters.
Stage 3: Use a TSP heuristic to sequence the stop points in each cluster, and calculate the total tour length for each cluster.
Stage 4: Return the smallest total tour length to the Genetic Algorithm.

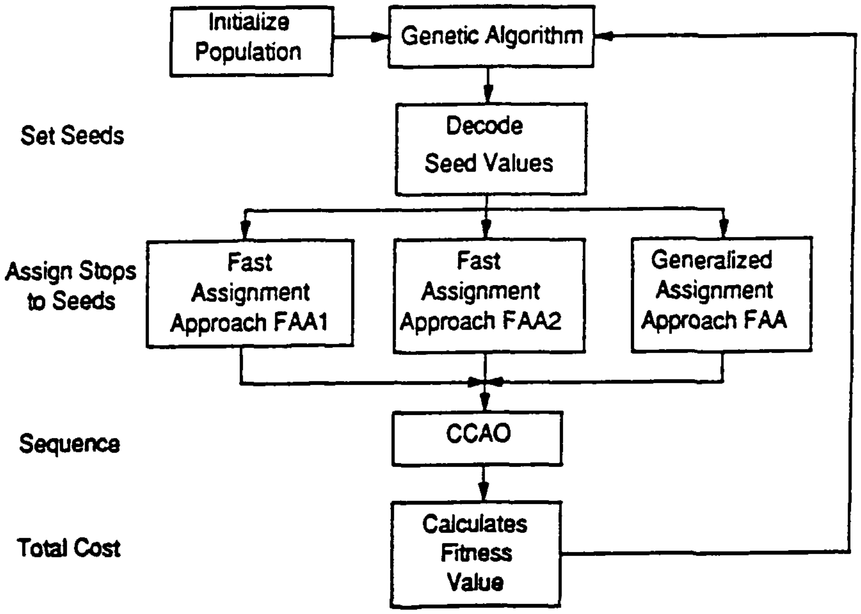


Figure 7: Fast Assignment Approaches FAA1, FAA2 and FGAA, are clustering algorithms. The same genetic material is shared between the three methods. The best solution is returned to the GA.

2.2.4.2.1 Clustering Method 1 [FAA1]: In this method, only one seed-point is active at a time. The nearest stop is assigned to the active seed-point, if doing so does not violate the corresponding constraint on vehicle capacity. For each stop assigned, a weighted distance factor is added to the active seed-point. The seed-point with the minimum weighted distance is made active for the next assignment. This process continues until all the stop points are assigned to some seed-point. The first method FAA1 uses the demand of a stop point (demand$_j$) as a penalty. The seed-point with the minimum penalty is chosen for the next assignment. This process goes on until all the stop points are assigned to the seed-point, in effect, until the clustering of the stop points is complete. Now a TSP heuristic, CCAO, is used to sequence the stop points. The algorithm checks for the capacity constraint of each of the vehicles in set K for the set of stops J. Each vehicle k in set K has an available capacity b_k, and the assignment of a stop j in set J to vehicle k consumes r_{kj} units of this capacity. A cost coefficient c_{kj} is a measure of the desirability of assigning stop j to vehicle k where $0 \leq \alpha \geq 1.0$ is the load factor (the portion of vehicle capacity to be filled). The following steps are used to assign the stops to a vehicle.

Step 1: Choose any seed-point S_k as the active seed-point.
Step 2: Assign stop j with the smallest cost coefficient c_{kj} to the active seed-point S_k.
c_{kj} = DISTANCE(stop j, seed k).

Step 3: Assign a weighted distance factor W_S to the active seed-point S_k.

$$W_s = \min_k (c_{kj}) + (\alpha * \text{demand}_j)$$

Step 4: If S_k has reached its full capacity, stop any further assignments to S_k.

Step 5: Activate the seed-point with the smallest W_{S_k} value.

Step 6: Repeat Step 2 through Step 5 until all stop points are assigned to one of the seed-points S_k.

2.2.4.2.2 Clustering Method 2 [FAA2]: The second method FAA2 uses a simple heuristic to do the clustering. Here all the seed-points are actively in the contest for receiving the next stop point assignment. The stop point with the minimum distance to any of the seed-points is selected and assigned to that seed-point. Clusters produced in the clustering step are fundamentally dependent on the locations of the seed-points. The role of GA in FAA2 is to use the fitness value to search for seed-points which the FAA2 can use to produce an assignment. The Fast Assignment Approach (FAA2) is a new algorithm, that is relatively fast and well suited for the genetic search because it uses a simple heuristic to do the clustering. To begin, each stop point j is given a weighted distance ranking. The stop point with the minimum distance to any of the seed points is selected and assigned to that seed point. The process is continued until the clustering of the stop points is complete. Now the CCAO TSP heuristic (Golden and Stewart, 1985) is used to sequence the stop points and the FAA2 algorithm checks for the capacity constraint of each of the vehicles. For each set of vehicles K and set of stops J each vehicle k in set K has an available capacity bk, and the assignment of a stop j in set J to vehicle k consumes rkj units of this capacity. A cost coefficient ckj is a measure of the desirability of assigning stop j to vehicle k.

The following steps are used to assign the stops to specific vehicles.

Step 1: Choose any seed-point as the active seed-point

Step 2: Assign the stop point *j* with the smallest cost-coefficient c_{kj} to the active seed-point Sk.

$$c_{kj} = \text{Distance (stop j, seed k)}.$$

Step 3: If S_k has reached its full capacity, stop any further assignments to Sk.

Step 4: Repeat Step 2 and Step 3 until all stop points are assigned to one of the seed points S_k.

2.2.4.2.3 Clustering Method 3 [FGAA]: A generalized assignment problem is solved to assign the stops to the seedpoints (Fisher and Jaikumar, 1981). A genetic search approach is used to extend the generalized assignment approach to routing problems with multiple vehicles constrained by capacity and stops with known demands for service. The general description of the three rudimentary steps of the algorithms follows:

Step 1. Calculate a "seed" location for each vehicle. The seeds provide nominal models of the directions and distances from a depot that the vehicles will travel.

Step 2. Using the seeds to set parameters, solve a generalized assignment mathematical model to obtain assignments of stops to vehicles.

Step 3. For each set of stops assigned to a vehicle, use an algorithm to calculate a traveling salesman tour, thus yielding a final solution to the problem.

A methodology has been developed that uses genetic search to find seed locations that tend to provide the generalized assignment model with parameters that consistently yield stop assignments that produce extremely efficient tours. For each set of vehicles K and a set of stops J each vehicle k in set K has an available capacity b_k and the assignment of a stop j in set J to vehicle k consumes r_{kj} units of this capacity. A cost coefficient c_{kj} is a measure of the desirability of assigning stop j to vehicle k. Given these parameters, the following generalized assignment problem (GAP) is solved to assign stops to vehicles.

$$\text{minimize} \sum_{k \in K} \sum_{j \in J} c_{kj} x_{kj}$$

subject to:

$$(1) \qquad \sum_{j \in J} r_{kj} x_{kj} \leq b_k \quad \text{for all } k \in K$$

$$(2) \quad \sum_{k \in K} x_{kj} = 1 \quad \text{for } all \ j \in J$$

$$x_{kj} = 0 \text{ or } 1 \qquad \text{for } all \ k \in K, \ j \in J.$$

The value of the decision variable, x_{kj}, is interpreted as follows:

$$x_{kj} = \begin{cases} 1 & \text{if stop j is assigned to vehicle k} \\ 0 & \text{otherwise} \end{cases}$$

Constraint set (2) forces each stop to be assigned to exactly one vehicle. Constraint set (1) limits the assignments by vehicle capacity.

A basic way to model a vehicle tour is to identify a single location called a seed, with the vehicle conceptually traveling from the depot to the seed and back. With this model, the extra distance incurred by adding stop j to the tour of vehicle k is given by

$$c_{kj} = \text{DISTANCE(depot, stop j)} + \text{DISTANCE(stop j, seed k)} - \text{DISTANCE(seed k, depot)}.$$

Given this definition of c_{kj}, the clusters produced in the Clustering Step are fundamentally dependent on the locations of the seed points.

2.2.4.2.4 Clustering Method 4 [COMBO method]: Many optimization problems require the investigation of multiple local optima. Here the concept of sharing functions (Goldberg 1987) is used to investigate the formation of stable subpopulations of different strings in the GA, thereby permitting the parallel search of many peaks. This method uses the string recommended by the GA on all three methods (FAA1, FAA2, FGAA) and the function with the best performance value is selected to return the fitness value for that particular string to the GA as shown in the Figure 7. The three methods (FAA1, FAA2, FGAA) all use the same string, and due to the competition between widely disparate points in the search space, help maintain a diverse population which searches many peaks in parallel. This multimodal optimization method also helps in avoiding premature convergence due to local optima

The following steps are involved in running the Combined method.

Step 1: Receive the string recommended by the GA.
Step 2: Evaluate function FAA1 using these strings.
Step 3: Evaluate function FAA2 using these strings.
Step 4: Evaluate function FGAA using these strings.
Step 5: Return the minimum of the three fitness values to the GA.

2.2.5 The Mechanics of the Adaptive Search

Let the complex process of multiple vehicle routing optimization be working in an environment E with a set of control parameters C which are available for the adaptive search strategy. Within each envir)nment there is a fitness measure for the performance of the VRP proces being developed using the *presently chosen control* parameters. In the routing problem, this fitness value is simply the agreed upon performance measure of the set of tours. It may be, for example, the total distance traveled for the whole fleet. Each environment e in E to which the controlled VRP process is subjected defines a performance response surface over the control parameter space C, defined by a fitness function F_e. It is the response surface defined by F_e that is explored by the adaptive search strategy in order to generate a good performance of the VRP process. In our problems, the function F_e is extremely complex, high-dimensional, multimodal and discontinuous.

As the genetic algorithm generate and test procedure generates better routing solutions, the GA exploits the accumulating knowledge of the VRP process being controlled. This is done by representing each point in the control parameter space as a binary string which encodes the performance character of a seed-point. In XVRP-GA, the control parameter is the location of the seed-point in 2 dimensional space. Each string has a field allocated for the performance function, F_e, which is returned after the string is evaluated by the evaluation function. The GA maintains a population of these control parameter strings as parent material for combining in a directed reproductive search to obtain more efficient tours. Each individual string is submitted for evaluation as a control parameter for the VRP process, and receives an associated performance measure from the evaluation function. Finally, using selection probabilities, these control parameter strings undergo reproduction with crossover and mutation genetic operators.

The population available for submission to GA operators is a collection of candidate control parameters C. Fixing one of the control parameters and leaving the other parameters free defines a hyperplane. Since each parameter has (1023 X 1023) possible locations we can have this many hyperplanes for each control parameter and there is a seperate control parameter for each available vehicle. When you consider all possible combinations of hyperplanes for all vehicles, it is obvious that *the search space is complex and multimodal. We observe from our experiments that* GA *rapidly exploits* accumulating information about F_e to restrict sampling to those hyperplanes which have a high expectation of good performance.

The search space defined by F_e is multimodal with relatively flat surfaces interspersed with spikes of good solutions. Because the search space is not a single gradient slope to a global minima, it is easy to become trapped in a local minima. GA avoids the penalty of local entrapment without the severe expense of simulated annealing by maintaining competing solutions employing different evaluation functions dispersed througout the search space. Because there are widely disparate points in the search space using multiple sharing evaluation functions, the tendency to prematurely converge on a less than best solution is minimized. FAA1, FAA2 and FGAA are all controlled by the same set of control parameters C, available form the adaptive GA. As the adaptive search progresses, each of the methods is likely to be sampling different hyperplanes looking for peaks in parallel. As good solutions are found and shared among the various competing methods the new "best" solutions will occasionally use the "best" control parameters discovered by the other functions to start searching a new hyperplane which has been shown by a competitor to provide good performance (just like a "me too" computer business which copies a successful one but puts its own twist on marketing).

Selection is the process of identifying the number of offspring each population member will bear. In nature, weak individuals tend to be less likely to survive to bear offspring. An analogous procedure is used in our population of candidate tours. In particular, 50 trials are carried out in each generation, and a population member is selected for parenting at each trial. The probability of any particular population member being selected in a trial is proportional to its relative fitness within the population as whole. Thus, poor performance tours are less likely to be selected to parent in the next generation than high performance tours. *The first few generations start with seed-point location values uniformly distributed* over the search space.

Figure 8(a) shows the seed-points produced by the COMBO method. Table 2 illustrates the method of encoding seed-point locations and generating offspring via the COMBO method. Two offspring strings are shown graphically in Figure 8(b) and (c). Table 2 illustrates the mechanics of a crossover genetic

Table 2. The Table Illustrates the Mechanics Of a Crossover Genetic Operator. The Seed String Shown In the Table Is the Actual Value That the COMBO Method Uses. The Encoded Version Of the Seed String Is Used by the GA in the COMBO Method. C11 and C12 are the Crossover Points In Parent 1 and C21 and C22 are the Crossover Points in Parent 2

The Mechanics of the Crossover Operator						
String	S_{x1}	S_{y1}	S_{x2}	S_{y2}	S_{x3}	S_{y3}
Seed-Points	100	100	900	100	900	900
Genetic Encoding	0001100100	0001100100	1110000100	0001100100	1110000100	1110000100
Parent 1	0001100100	0001100100	c_{11}1110000100	00011 c_{21} 00100	1110000100	1110000100
Parent 2	0001100100	c_{21} 0001100100	11100 c_{22} 00100	0001100100	1110000100	1110000100
Offspring 1	0001100100	0001100100	0001100100	1110000100	1110000100	1110000100
Decode 1	100	100	100	900	900	900
Offspring 2	0001100100	1110000100	0001100100	0001100100	1110000100	1110000100
Decode 2	100	900	100	100	900	900

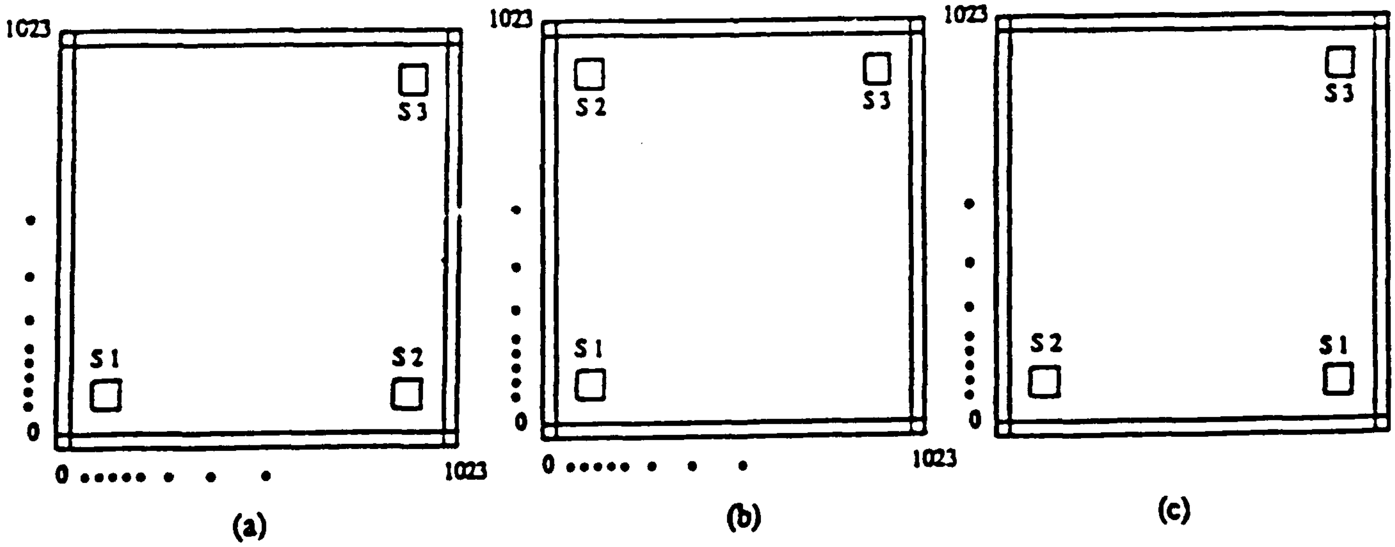


Figure 8(a), (b), (c). The Figure (a) Shows the Seed-Point Produced By the COMBO Method. The Two Offspring Strings Are Shown Graphically in Figures (b) and (c) operator with the decoded version shown for clarity. The encoded version of the string is used by the GA. To illustrate the effect of crossover, let us assume we are using the standard 2-point crossover method. C11 and C12 are the crossover points in Parent 1 and C21 and C22 are the crossover points in Parent 2. Now, if the genetic material between C11 and C12, C21 and C22 are exchanged, two offspring strings are generated. The use of the crossover has resulted in two candidate seed-point locations, with seed-point 3 not being effected by the crossover operation, but seed-point 1 and 2 are moved to a new location. These new locations of the seed-points results in different clusters, which in turn results in a different fitness measure (total distance). The stop locations are sequenced in each of the candidate clusters in a cost effective way (TSP) and the resuit returned to the GA. As generations progress, seed-points tend to be concentrated in tight geographical areas due to the survival of the fittest mechanism of the GA. This is illustrated in Figure 9(a) where the total run of 1000 trials is shown and in Figure 9(b) where only the last 50 trials are plotted. A trial is a single execution of the evaluation function on a candidate control parameter, and a generation consists of exhaustive trials on all candidate offspring.

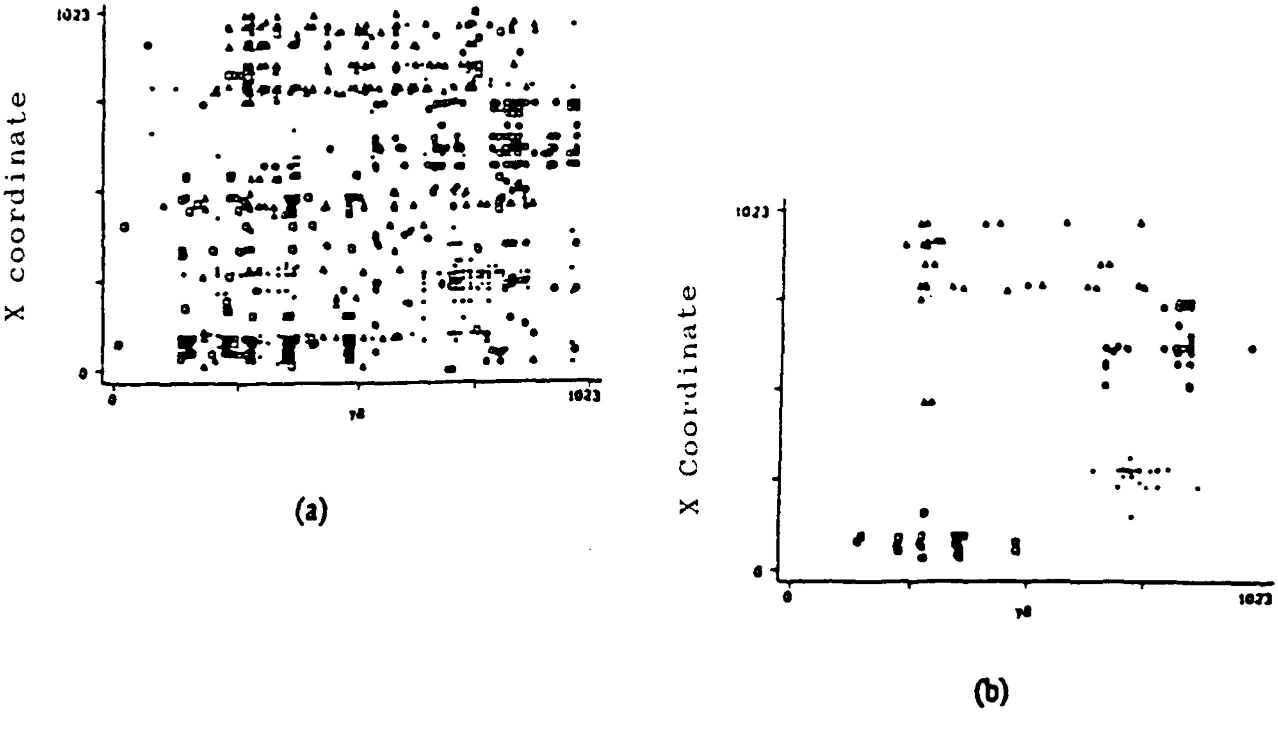


Figure 9: Seed-points tend to be concentrated in tight geographical areas. All the 1000 trials are plotted in (a) and in Figure 9(b) only the last 50 trials are plotted.

The three performance curves on the graph shown in Figure 10 illustrate the survival of the fittest nature of the GA search. The performance measure is total distance traveled by the fleet, so small values are desirable. In Figure 10, the top curve indicates tne worst performance of the evaluation function as function of generations. The bottom curve indicates the best performance in each generation. The middle curve is the plot of the average performance of the evaluation function. The decreasing trend in the curves illustrates the survival of the fittest candidates in the population, and indicates that the GA is doing much better than a random walk in the control parameter search space.

Table 3 presents empirical work that illustrates the parallel nature of the search on a four vehicle problem. The values shown in Table 3 are generated when each evaluation function (FAA1, FAA2, FGAA) finds a seedpoint parameter which produces a performance value better than the best found up to that point in time. The FAA1 method produces the first best solution (example 1). FAA2 then identifies a sequence of seven improving solutions, as shown in examples 2 through 8. The seed-points that produce these solutions are the result of searches centered around a few "good" spots. At generation 14, the GA produces a seed-point location that FAA1 adopts and improves. The coordinate values reveal that this seed-point location is essentially the one FAA2 was using to improve the solution performance, as shown in example 9. The FGAA method which had not generated a best solution in earlier generations, produces one at generation 17 using seed-points from generation 2. Also note that the seed-points in example 11 are in a completely different area of the search space. This illustrates the parallel search of a multimodal response surface occurring in the algorithm.

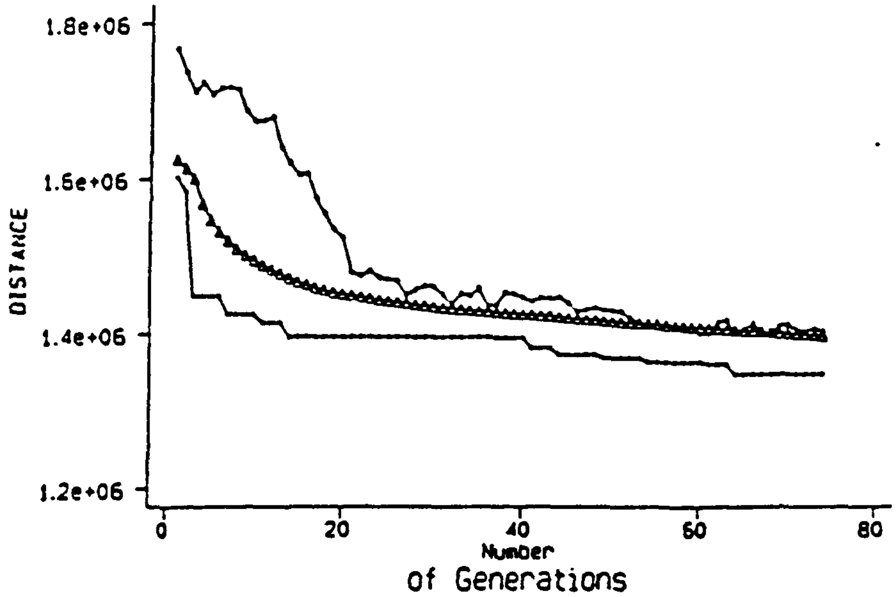


Figure 10: The top curve indicates the worst performance of the evaluation function as function of generations. The bottom curve indicates the best performance in each generation. The middle curve is the plot of the average performance of the evaluation function.

2.2.6 Functional Description of the Local Optimization Post-Processor XCHANGE

Although the results of the COMBO method were satisfactory, there were some questions as to local improvements which might be achieved in the solutions produced. To explore the possibility of improving the solutions by making small, intelligent swaps around the "good" solution and XCHANGE algorithm was developed. The "good" clusters are first determined using the COMBO method then the XCHANGE method is used as a simple postprocessor which derives its power from the use of a genetic algorithm. The XCHANGE method uses the genetic string to interpret the stop assignments of a particular route. The effect of the genetic recombinations is to make simple alterations to the existing system of routes. The method chooses outlying stops and investigates the effect of swapping the stop to make it a member of an adjacent route. The stop is offloaded onto another route only if it does not violate the capacity constraint for the vehicle serving that route. The procedure continues for a set number of trials.

Table 3: Parallel nature of the adaptive search. Each of the three methods is able to exploit promising seed-points locales discovered by the other methods.

Ex	Sx1	Sy1	Sx2	Sy2	Sx3	Sy3	Sx4	Sy4	Perf	Method	Generation
1	572	61	959	623	742	43	125	463	12373	FAA1	1
2	812	824	316	528	981	405	181	816	12082	FAA2	2
3	759	851	85	371	49	136	968	279	11880	FAA2	2
4	580	928	105	396	963	818	82	275	11665	FAA2	2
5	466	716	805	75	114	769	494	873	11518	FAA2	2
6	279	488	865	65	51	747	944	660	11132	FAA2	2
7	232	791	865	79	901	672	197	720	10958	FAA2	2
8	757	329	110	833	55	136	968	663	10761	FAA2	7
9	714	182	90	841	52	141	976	652	10732	FAA1	14
10	714	342	105	833	55	128	1006	648	10606	FAA2	16
11	232	151	873	185	53	795	958	464	10490	FGAA	17
12	232	150	873	185	54	868	958	464	10450	FGAA	30
13	773	181	150	185	55	868	945	431	10394	FGAA	32
14	688	197	118	185	55	868	977	524	10373	FGAA	35
15	693	169	83	838	72	151	943	908	10299	FAA2	38

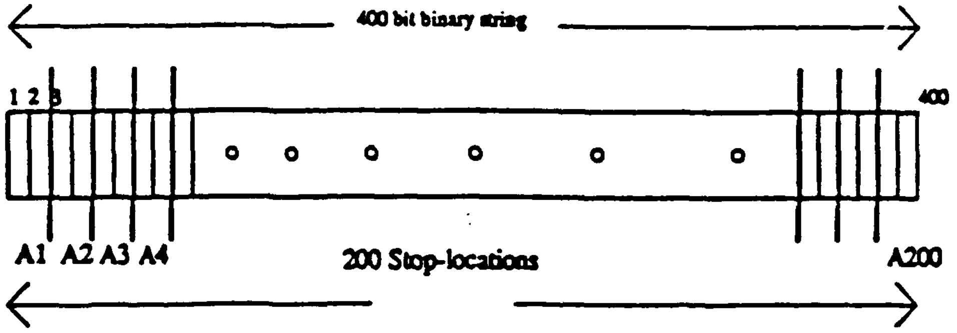


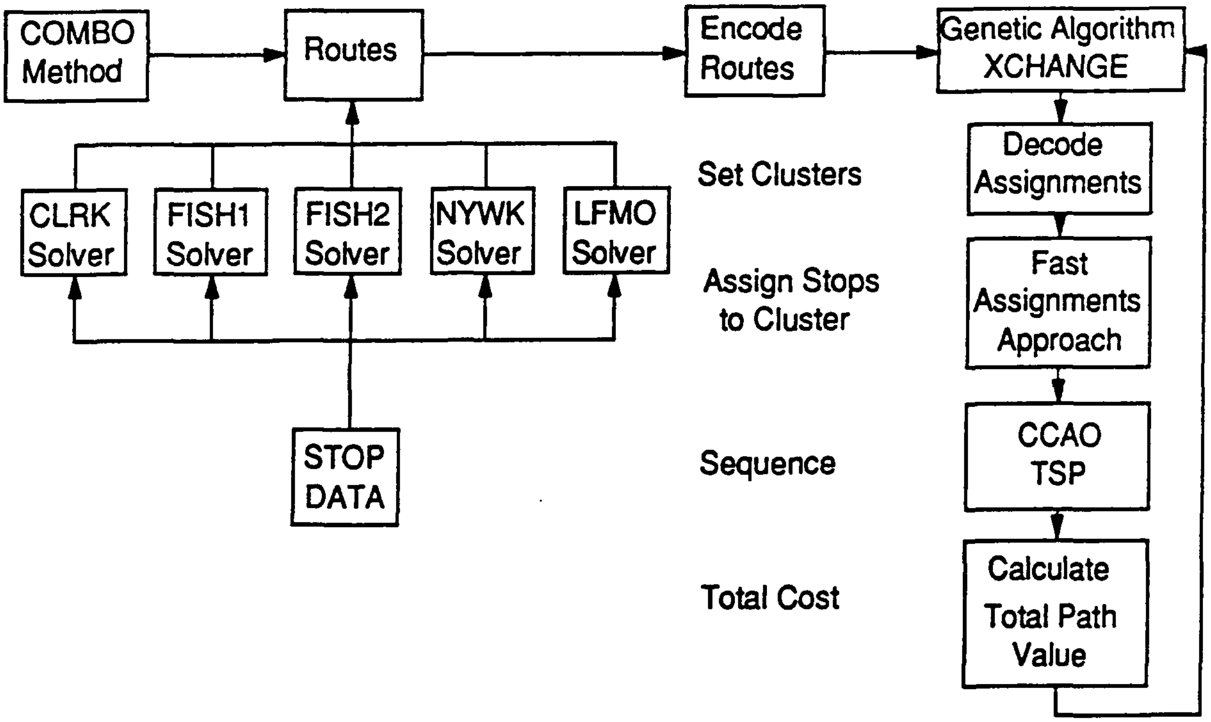
Figure 11: Each stop-point is identified by a cluster number which is represented by a 2 bit binary string.

2.2.6.1 Representation

The genetic algorithm in XCHANGE use bit strings as chromosonal encodings of the problem they are trying to solve. Basically, candidate solutions to the problem are represented as bit strings (chromosomes), and populations of these solutions are simulated over some number of generations. The control parameters are vehicle assignments. The assignments represent a good cluster which the truck serves. The number of clusters recommended by the GA is equal to the number of vehicles available. Each stop point is identified by a cluster number which is represented by a 3 bit binary string to represent the numbers 0 to 7 inclusive for 8 vehicles. If only four vehicles are used the cluster can be represented by a 2 bit binary string as shown in Figure 9. The binary bits are rescaled if there is an odd number of clusters. The length of the GA string is calculated using the formula $L_{GA} = \sum_{j=1}^{N} L_j$, where L_j is the length of one parameter string.

2.2.6.2 The Evaluation Functions.

The evaluation functions are applied to the stop-point clusters of an existing route produced by the COMBO method to initiate the process. We then sequence the stop locations in a cost effective wav using a TSP for sequencing. In the TSP, a tour begins at a home location, visits each stop on a list exactly once, then returns to the location of origin. The objective is to find the order in which the stops are visited so that the total distance traveled is as small as possible. The algorithm checks for the capacity constraint of each of the vehicles. There is a set of vehicles K and a set of stops J. Each vehicle k in set K has an available capacity b_k, and the assignment of a stop j in set J to vehicle k consumes r_{ij} units of this capacity.



STOP
DATA

Total Cost

Figure 12: The Figure illustrates the overall architecture of the XCHANGE method. The result derived from the COMBO, CLRK, FISH1, FISH2, NYWK and LFMO methods are encoded and used as the initial cluster information.

Stage 1: Accept the stop assignments (clusters) recommended by the Genetic Algorithm.
Stage 2: Use a TSP heuristic to sequence the stop points in each cluster, and calculate the total tour length for each cluster.
Stage 3: Assign distance value to each cluster A_k.
Stage 4: If A_k has exceeded its full capacity (Vehicle capacity), reject this recommendation and return a large penalty value to the genetic algorithm.
Stage 5: If it does not violate the capacity constraint return the total tour length to the Genetic Algorithm.

Figure 12 illustrates the overall architecture of the XCHANGE method. The result derived from the COMBO method is encoded and used as the initial cluster information. The GA uses this information and applies the genetic recombination operators to improve the route distance. In separate experiments, the results derived from the five models (CLRK, FISH1, FISH2, NYWK and LFMO) are used as cluster information in the initial population of the genetic algorithm. This shows one can use the XCHANGE method as a post-processor for any of the VRP solvers.

2.2.6.3 The Mechanics of the Adaptive Search

If we let C represent the set of control parameters available for the adaptive search strategy then we can define within each environment a fitness measure for the performance of the VRP process under the control parameters. As "good" solutions are generated the GA uses the behavior of the search space and exploits the accumulating knowledge of the VRP process being controlled to generate better solutions. Each point in the control parameter space is represented as a binary genetic string which is operated on by the genetic operators, evaluated for efficiency by the evaluation function, and then subjected to the survival of the fittest rule. A post processor, XCHANGE, then attempts to improve on the route efficiency by swapping points among neighboring routes. Each string in a stop location has a field allocated for the performance function Fe, which is returned after being quantified by the evaluation function.

As we have previously noted, the population we are searching is a collection of candidate control parameters C and fixing one of the control parameters leaves the other parameters free to define a hyperplane. Each parameter has 0 to N possible locations where N is the number of vehicles. The main aspect to note here is that the mutation rate is kept quite low (0.00001) and used sparingly to avoid disruption of the clusters and to allow for convergence to good solutions. As shown in Figure 13, the crossover operation does not create nearly as much disruption in the clusters as mutation and is the main method to search the solution space for good routing clusters. Due to the immense number of suboptima and small difference in solution length between respective clusters, the GA converges very slowly and many generations are required to find good tours. It is important to note that our efficient algorithms are of such quality that they generate new generations of offspring and evaluate them very quickly so as to suit the algorithms to run in real time applications.

Figure 13 illustrates the mechanics of a crossover genetic operator. Figure 13 (a) shows the routes produced by the COMBO method. The assignment string shown in Table 4 is the decoded value that the XCHANGE method uses. The encoded version of the assignment string is used by the GA in XCHANGE method. To illustrate the effect of crossover, let us assume we are using the standard 2-point crossover method. C11 and C12 are the crossover points in Parent 1 and C21 and C22 are the crossover points in Parent 2. Now, if the genetic material between C11 and C12, C21 and C22 are exchanged, two offspring strings are generated. These two offspring strings are shown graphically in Figures 13(b) and (c). The use of the crossover has resulted in two candidate clusters, with stop 9 being transferred from cluster 2 to cluster 0, as shown in Figure 13(b) and stop 11 being transferred from cluster 2 to cluster 3, as shown in Figure 13(c). We then sequence the stop locations in each of the candidate clusters in a cost effective way and return the result to the GA.

Table 4. The Table Illustrates the Mechanics of a Crossover Genetic Operator. The Assignment String Shown in the Table is the Actual Value That the XCHANGE Method Uses. The Encoded Version of the Assignment String Is Used By the GA in the XCHANGE Method. C11 and C12 Are the Crossover Points In Parent 1 and C21 and C22 are the Crossover Point in Parent 2

The Mechanics of the Crossover Operator																
String	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Assignments	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
Genetic Encoding	00	00	00	00	01	01	01	01	10	10	10	10	11	11	11	11
Parent 1	00	00	00	00	01	0_{c11} 1	01	01	1_{c21} 0	10	10	10	11	11	11	11
Parent 2	00	00	00	00	01	01	01	01	$_{c21}$ 10	10	1 0_{c22}	10	11	11	11	11
Offspring 1	00	00	00	00	01	01	01	01	00	10	10	10	11	11	11	11
Offspring 2	00	00	00	00	01	01	01	01	10	10	11	10	11	11	11	11

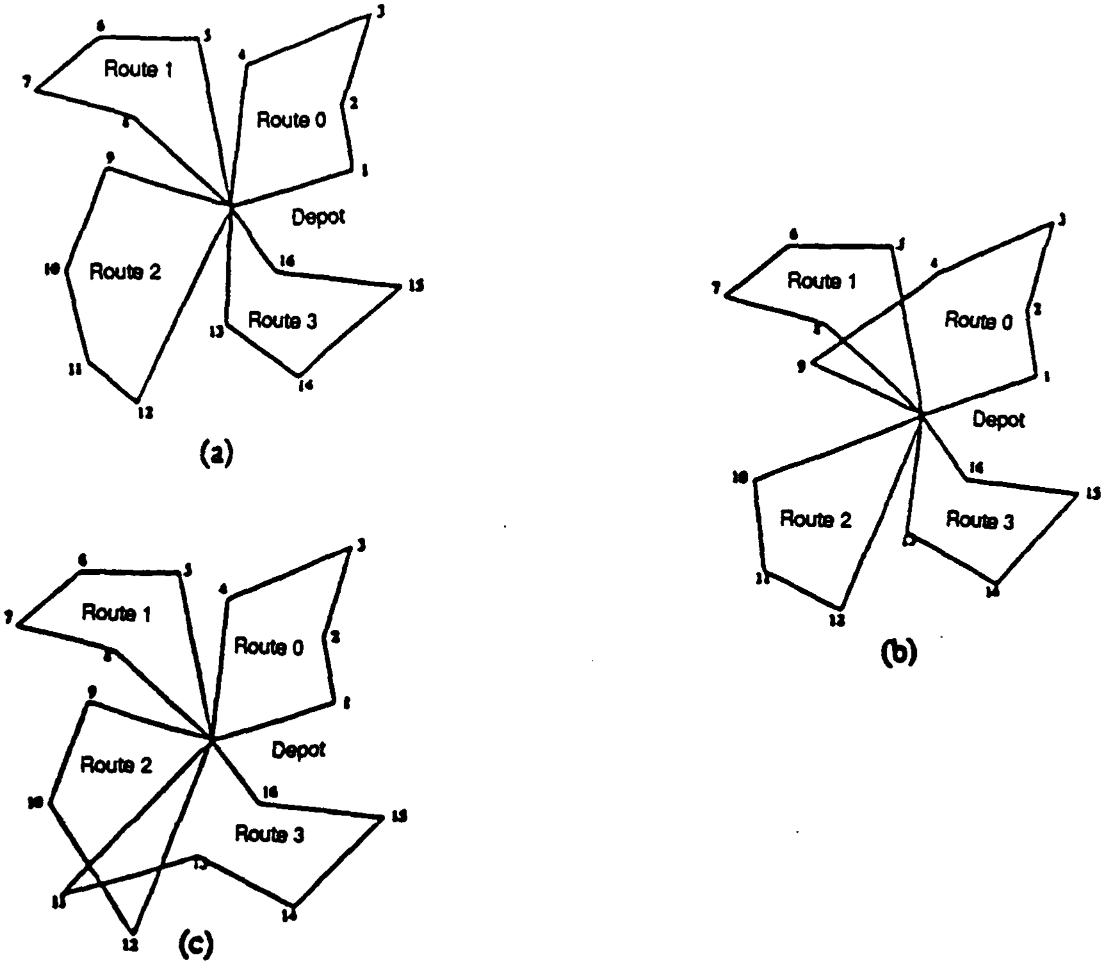
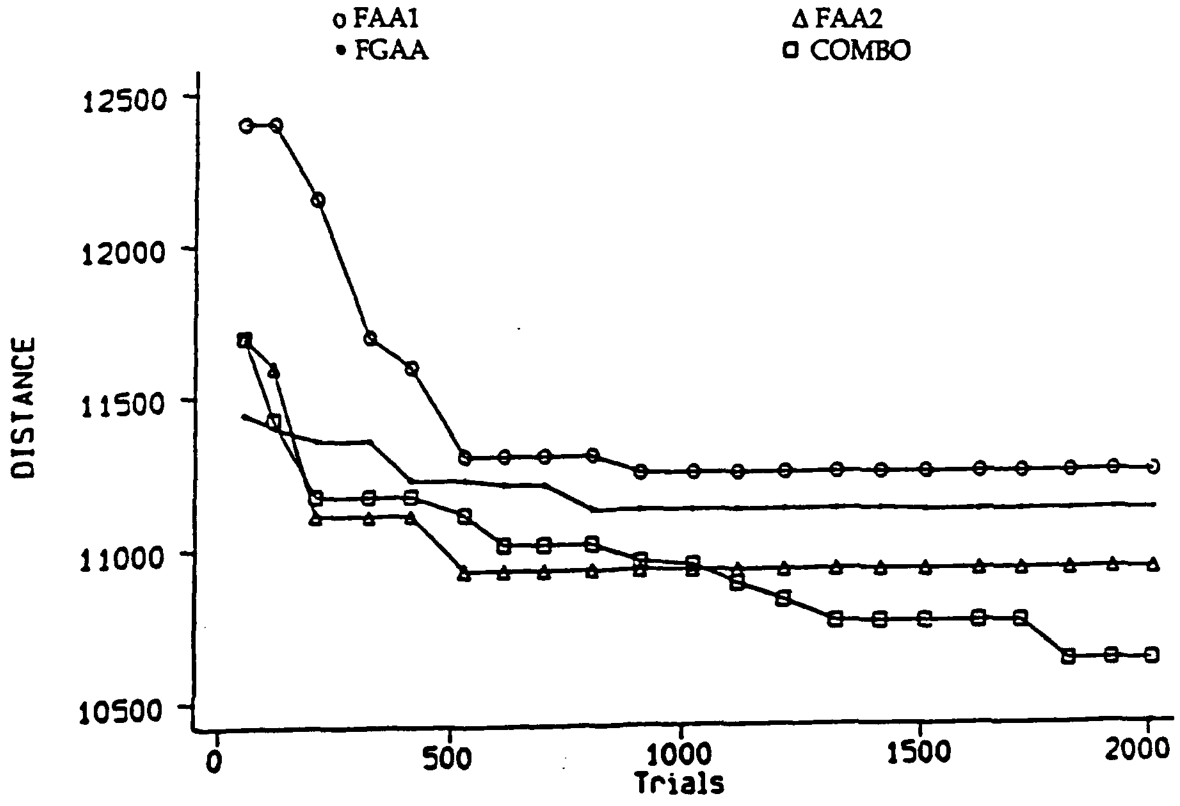


Figure 13. The Figure Illustrates the Mechanics of a Crossover Genetic Operator. The Routes Produced By the XCHANGE Method is Shown in (a). The Two Offspring Strings Are Shown Graphically in Figures (b) and (c)

2.3 Experimental Results

To evaluate solution quality, the problems were solved with five different VRP solvers as well as XVRP. For each problem, each of the 5 methods that were used in the work described were available and run in contrast with the new genetic algorithm method (XVRP). The five alternative algorithms are:

- Clarke-Wright, a venerable heuristic algorithm capable of producing fast solutions to large scale problems with multiple vehicles and an objective of minimizing total distance (Clarke & Wright, 1964)
- FISHER1, a heuristic for the multiple vehicle problem that uses a generalized assignment model and produces high-quality solutions to small and medium size problems (Fisher & Jaikumar, 1981)
- FISHER2, a heuristic for the multiple vehicle problem that uses a generalized assignment model and uses a different seed setting strategy (Nelson, 1983)
- NYGARD-WALKER, a heuristic that uses space-filling curves and Lagrangian relaxation to obtain solutions to very large-scale multiple vehicle problems (Nygard & Walker, 1988).
- LFMO, a vehicle routing solver based on Spanning Trees and Branch Exchanges (Hongjun Liu, 1989).



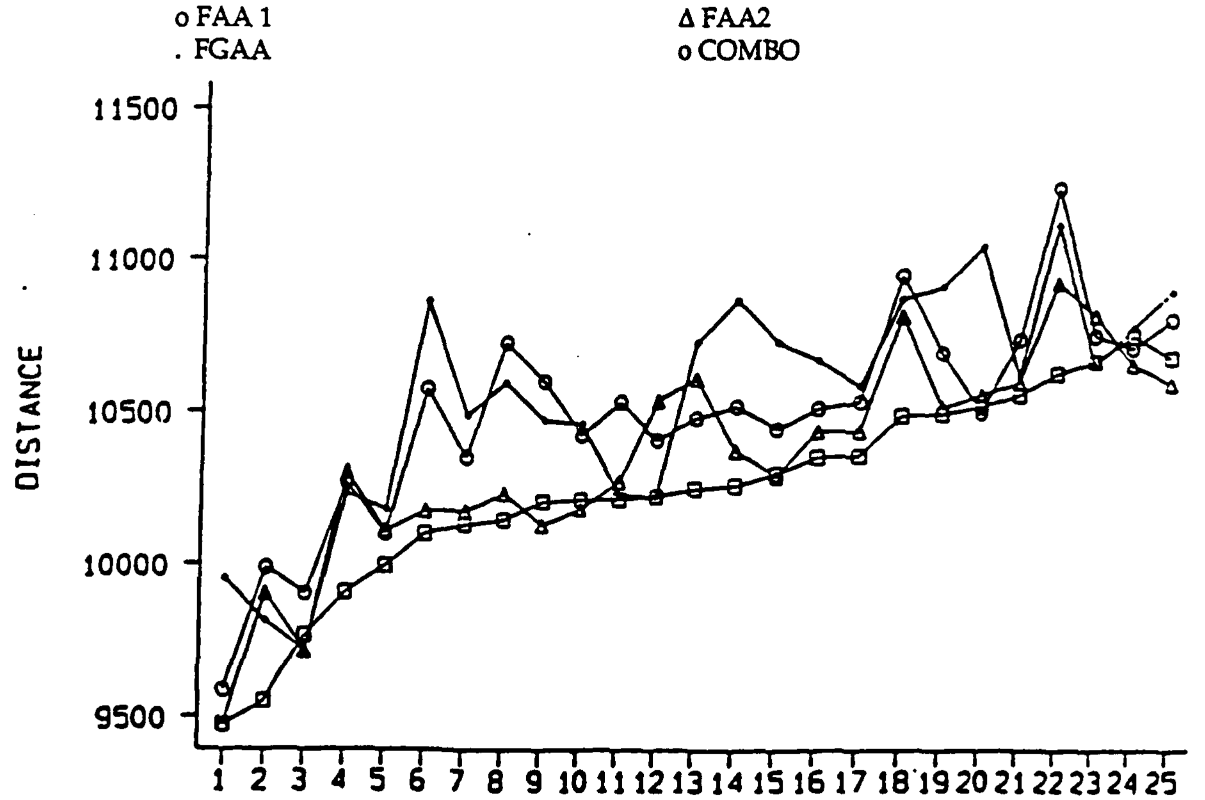
COMBO Vs other Methods

Figure 14. Comparison of performance of FAA1, FAA2, FGAA and COMBO methods for a single problem. The best solution in 1000 trials is plotted.

To evaluate the effectiveness of the XVRP system, a random problem generator was used (Nelson, 1983). The test problems are fully dense with 100, 200, and 1000 stop points, use vehicles with a utilization factor of 95%, and are generated in a square, 1023 miles on each side. Performance is based on the final quality of the total distance traveled by all the vehicles. All experiments were performed using a modified GENESIS (Grefenstette, 1984) system. A mutation rate of .001 was set. This means that each bit in the representation of each population member is changed (from O to 1 or vice versa) with a probability of .001. The experiments were set for 1000 trials of solutions. In all cases, a selection/insertion heuristic called CCAO [Golden and Stewart 1985], was used to calculate the traveling salesman tours. The best solution found in 1000 trials was retained. A considerable number of experiments were run on a network of SUN 3 workstations. The results demonstrate considerable improvement in the GA search by using multiple sharing evaluation functions. When this method is used, relatively powerful genetic search can be conducted for parameter discovery in an environment of networked desktop workstations.

2.3.1 Performance With Multiple Sharing Evaluation Functions Using COMBO Method.

In order to show the performance improvements of the COMBO method, each of the 25 problems were run for 2000 trials using the FAA1, FAA2 and FGAA methods. All the GA parameters were kept constant throughout the experiments. The test problems were fully dense with 200 stop points, 4 vehicles with a utilization factor of 95%. The performance improvement of the GA by using multiple sharing evaluation functions (COMBO method) for a single problem is illustrated in Figure 14. Note the three



COMBO Vs other Methods

Figure 15: Multiple evaluation functions results. The COMBO method consistently performs better than that achieved using any single evaluation function

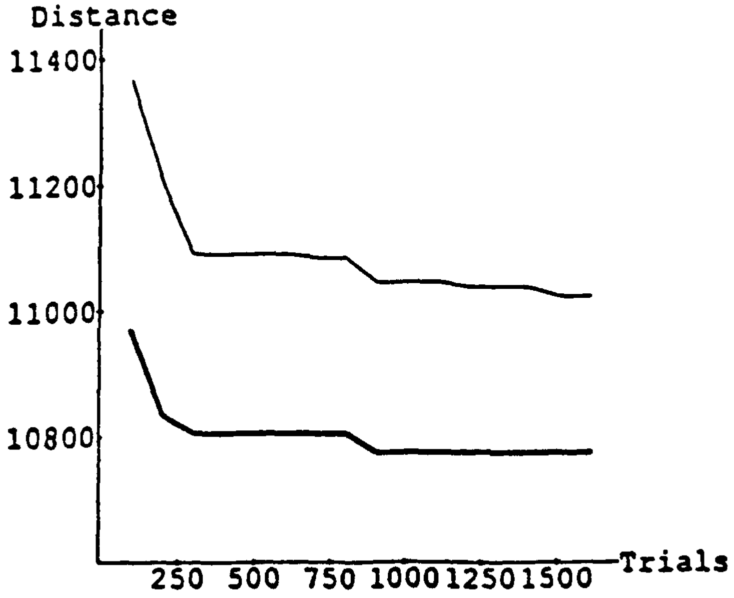


Figure 16: The best assignment achieved from the COMBO method is used as a initial starting point for the genetic search. The performance curve of the XCHANGE method continues to drop through subsequent trials.

individual methods do not improve the search after about 1000 trials, but the performance curve of the COMBO method continues to drop through subsequent trials. This result is consistent over all 25 problems tested, and is indicated in Figure 15.

2.3.2 Performance Improvement With the Local Optimization Post Processor Using XCHANGE Method

In order to show the performance improvements of the XCHANGE method, each of the 25 problems were run for 1000 trials using the genetic algorithm to improve the assignments. The test problems were fully dense with 200 stop points, 4 vehicles with a utilization factor of 95%. All the GA parameters, such as crossover rate, mutation rate, population size, and string length were kept constant throughout the experiments. The performance improvement of the GA by using assignment evaluation function for a single problem is illustrated in Figure 16.

Note the XCHANGE method is a post-processor module. The best assignment achieved from the COMBO method is used as an initial starting point for the genetic search. The performance curve of the XCHANGE method continues to drop through subsequent trials. This result is consistent over most of the other 25 problems tested, and is indicated in Figure 17. A summary of the computational results is shown in Table 5. It illustrates the performance measure of the various models used for benchmark testing of the XVRP system. The values shown in the table are the total miles traveled using four vehicles. Further, as shown, each model behaves differently with each data set, and XVRP performs better in each case. This is due to the adaptive capability of XVRP.

Table 5: The Table illustrates the performance measure of the various models used for benchmark testing of the XVRP-GA system. The values shown in the table are the total miles traveled using four vehicles. The performance improvement of the GA by using multiple sharing evaluation functions (COMBO method) is illustrated. The XVRP-GA system did perform better than all the other Algorithms. The values shown are the actual performance of the different algorithms in miles.

200 Node • 4 Vehicle • 95% Utilization

PROBLEMS	CIRIC	FISH1	FISH2	NYWK	LFMO	FAA1	FAA2	FGAA	COMBO	XCHANGE
p1.1	11501	10887	11123	11025	11340	10755	10823	10656	10670	10600
p1.2	11522	11251	10943	11418	11003	10532	10273	10230	10215	10163
p1.3	11345	10521	10567	11150	10834	10484	10611	10732	10252	10248
p1.4	11174	10976	10948	12046	10636	10423	10182	10461	10212	10129
p1.5	10950	11307	10863	10552	10657	10726	10233	10553	10145	10021
p1.6	11807	11306	12357	10756	11285	11247	10931	11121	10629	10629
p1.7	10936	10792	10796	10643	10614	10447	10290	10731	10255	10245
p1.8	10556	10716	10758	10572	10356	10098	10108	10178	9994	9905
p1.9	11542	10616	11234	11525	10825	10267	10301	10231	9904	9904
p1.10	10003	10829	10879	10056	10532	9584	9472	9950	9472	9472
p1.11	11023	10621	11014	10845	10509	10350	10176	10489	10130	10125
p1.12	11501	10841	10827	12053	10602	10513	10440	10670	10354	10313
p1.13	11883	11211	11253	11028	11127	10697	10518	10916	10498	10498
p1.14	11254	10961	11113	11763	11425	10501	10558	11047	10521	10519
p1.15	11187	10441	10567	10982	10950	10572	10175	10860	10100	10100
p1.16	11058	11363	11363	11187	11370	10810	10596	10902	10884	10790
p1.17	11665	11250	11250	10968	10934	10520	10375	10870	10261	10200
p1.18	11114	10189	10190	10481	10778	9986	9900	9810	9549	9475
p1.19	11448	10643	11002	10360	10649	10597	10129	10472	10207	10135
p1.20	12538	10885	11063	11846	11686	10715	10660	10784	10755	10755
p1.21	11883	10938	11124	11060	10468	10408	10539	10225	10225	10225
p1.22	10963	10586	10494	10174	10174	9900	9713	9713	9765	9765
p1.23	11212	10638	10776	11237	10632	10742	10599	10619	10558	10558
p1.24	11778	10967	11216	11412	11481	10952	10818	10875	10491	10491
p1.25	11570	10636	10595	11186	10766	10535	10440	10585	10355	10355

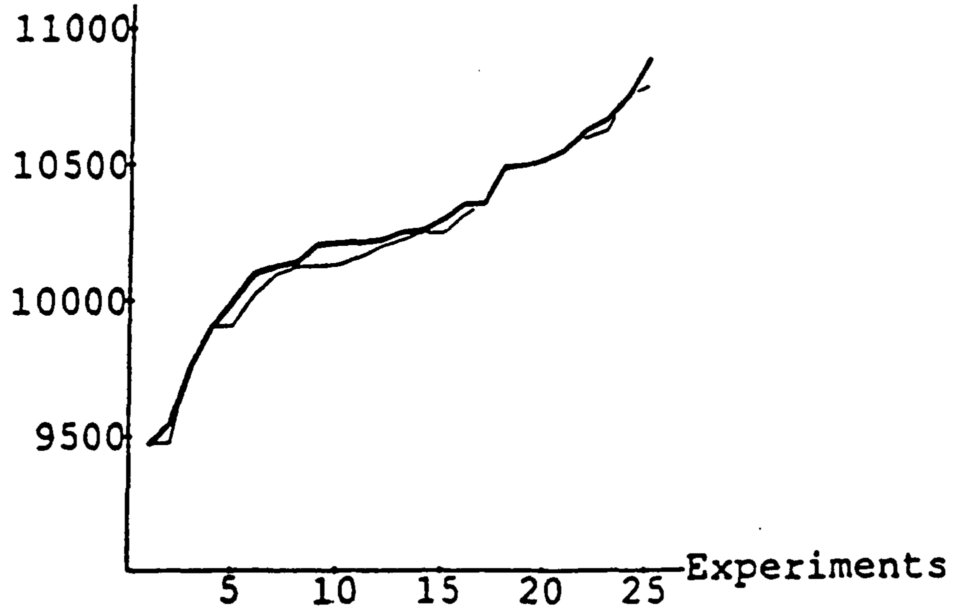


Figure 17: The performance curve of the XCHANGE method is seen to improve. This result is consistent over most of the other 25 problems tested.

The XCHANGE method was also run on the solutions produced by the five routing algorithms. The results are shown in Table 6. The assignment achieved from the various algorithms is used as an initial starting point for the genetic search. In most cases the XCHANGE post-processor improved the routes by up to 9%.

Table 6: The values shown are the actual performance of the different algorithms in miles. In most cases XCHANGE produced the better solution than the ones produced by the various algorithms. In each case the XCHANGE method started with routes (clusters) produced by the various algorithms. The exchange values for each problem are derived by injecting the initial population with routes generated by each solver. The BEST column indicates the performances improvements of XCHANGE when the initial population is injected with the routes (clusters) of all five solvers.

PRBM	CLRK	FISH1	FISH2	NYWK	LFMO	BEST
p1.1	8897	8590	8742	8668	8955	
xchnge	8831	8589	8742	8610	8706	8589
p1.2	8985	8824	8579	8938	8479	
xchnge	8325	8824	8579	8780	8329	8325
p1.3	8897	8680	8609	8872	8593	
xchnge	8548	8664	8593	8825	8593	8417
p1.4	8826	8648	8663	9645	8392	
xchnge	8348	8648	8663	9188	8392	8250
p1.5	8739	8849	8491	8605	8373	
xchnge	8503	8848	8473	8198	8148	8148
p1.6	9249	8800	9643	8510	8897	
xchnge	8915	8799	9643	8510	8788	8510
p1.7	8639	8714	8624	8574	8391	
xchnge	8115	8658	8568	8443	8391	8115
p1.8	8363	8251	8243	8467	8034	
xchnge	8272	8250	8242	8467	7900	7796
p1.9	8937	8281	8730	9038	8574	
xchnge	8481	8280	8730	9038	8401	8071
p1.10	7941	8676	8564	7987	8290	
xchnge	7935	8675	8522	7885	8290	7801

2.3.3 Statistical Comparison of Routing Algorithms

The results shown in Table 7 is used to illustrate statistical techniques in choosing alternative heuristic algorithms (Golden B.L., Stewart, W.R., 1985) (Yang, 1990). In order to verify if solutions from the various heuristics solutions values are normally distributed, we computed the Kolmogorov-Smirnov statistics. The Kolmogorov-Smirnov method is used to test whether the distribution function is a normal distribution when the sample size is between 20 and 30. We use the XCHANGE, COMBO, FAA2 and FGAA methods for comparison. Since the Kolmogorov-Smirnov method indicates that the distribution of the data is not normal, the Friedman Test is used.

The Friedman test is used to compare three or more different heuristic algorithms at a time if the hypothesis that the data comes from a normal population is rejected. The Friedman test is a nonparametric counterpart of the parametric two-way analysis of variance (ANOVA) test and is used to test whether three or more different heuristic algorithms mean costs are equal. The test may also be used if only random data are available. The data are set in a randomized block design with n problems each containing k algorithms. The meast rements are ranked in each problem over the algorithm. When this has been done for each p-oblem, the ranks are summed for each algorithm. In the case of ties, average ranks are used. The null hypothesis is that all the algorithms (three or more) have equal mean costs and the alternative hypothesis is that all the algorithms (three or more) do not have equal mean costs. Using the multiple comparison test, if the rank sums of any two algorithms are greater than 12.06 units apart, they may be regarded as unequal. Therefore, it can be concluded that the XCHANGE algorithm may be regarded as superior to the COMBO algorithm and the COMBO algorithm may be regarded as superior to the FAA2 algorithm and the FAA2 algorithm may be regarded as superior to the FGAA algorithm. In order to compare three or more heuristic algorithms, the expected utility approach was selected. The following table (Table 7) illustrates the fact that the XCHANGE algorithm appears to be the most accurate of the all heuristic algorithms.

Table 7: Comparison of the Accuracy of Three Heuristics Using Expected Utility

	Sample	Standard		Expected
Algorithm Name	Mean	Deviation	b value	c value
XCHANGE	10.23	0.34	0.01	898.62
COMBO	10.26	0.34	0.01	920.26
FAA2	10.35	0.33	0.01	960.04
FGAA	10.55	0.37	0.01	827.17

Note:
b value = sample variance / sample mean.
c value = sample mean / b value.
A utility function where alpha = 600, beta = 100, and t = 0.05 was selected.

2.3.4 *Computational Testing of the XVRP System*

To illustrate the low memory requirements of XVRP for large routing problems, test problems were generated with 1000 stop points and 4 vehicles were used with a utilization factor of 90%. The genetic algorithm requires little memory and can achieve good solutions within seconds, but can require substantial computer time (perhaps several hours) to achieve top quality solutions. However, the genetic algorithm lends itself naturally to asynchronous parallelization on a network of workstations (Kadaba, N., Nygard, K. 1990) (Kadaba, *et al.* 1990).

Table 8 shows the memory requirements for Clarke-Wright model and the XVRP system solving a 1000 node problem. The statistics were gathered separately on a SUN 3/260 with 8 Meg RAM and a (SUN 4) Solbourne 5/802 with 48 Meg of RAM, 4 Gigabytes of disk, 2 SPARC CPU's at 33 MHz for a total MIPS of 37 and total MFLOPS of 7. Table 9 displays information about Clarke-Wright and XVRP running under a UNIX operating system. Table 8 compares the memory requirements of Clarke-Wright and XVRP. The table displays the process ID under PID and the control terminal identifier under TT. The combined size of the data and stack segments (in kilobyte units) is shown under SIZE. The real memory (resident set) size of the process is shown under RSS. %CPU and %MEM display the percent CPU utilization and percent of real memory used by the two processes at that particular instant.

The total memory requirements for the Clarke-Wright process is around 6 Meg, whereas it is 1 Meg for the XVRP process. Note, the Clarke-Wright process is getting a very small usage of the CPU on the SUN 3/260. This is due to the fact that the Clarke-Wright process is spending all the time in disk I/O for swapping the huge virtual memory. The low memory requirements and relative speed of XVRP sets the stage for networks of microcomputers to be brought to use on very large routing and scheduling problems. The good performance XVRP method is consistent over most of the other 10 problems tested, and is indicated in Table 6.

Table 8: The Table shows the Memory Requirements for Clarke-Wright model and the XVRP-GAs system solving a 1000 node problem. The statistics were gathered separately on a SUN 3/260 and a Dual SPARC CPU SUN 4. The description of the Table headings is explained in the paper.

SUN 3-260 with 8 Meg Ram

PID	TT	SIZE	RSS	%CPU	%MEM	COMMAND
14809	p0	5960	3792	0.8	52.1	Clarke-Wright
14899	p0	1158	980	46.4	5.3	xvrp-ga

SUN 4 (Solbourne) with 48 Meg Ram and Dual SPARC CPU

PID	TT	SIZE	RSS	%CPU	%MEM	COMMAND	CPU
24536	pb	5936	5128	51.6	13.9	Clarke-Wright	CPU 1
10100	pe	1160	992	80.2	2.7	xvrp-ga	CPU 2

Table 9: The Table illustrates the performance measure of the Clarke-Wright model used for benchmark testing of the XVRP-GA system. The values shown in the Table are the total miles traveled using four vehicles. The good performance XVRP-GA method is consistent over most of the 10 problems tested, and is indicated in Table. The values shown are the actual performance of the different algorithms in miles.

1000 Node Problems

PRB	CLRK	TIME	XVRP-GA	TIME	% IMPROVE
p1.1	27544	8220	24449	5037	11.9
p1.2	27386	5700	24808	8146	9.9
p1.3	26968	4800	24692	2383	8.8
p1.4	26785	5400	24628	1377	8.4
p1.5	26506	6720	24461	1368	8.0
p1.6	26974	6720	24716	1419	8.7
p1.7	27083	4620	24747	1854	9.0
p1.8	26935	4560	25061	1389	7.2
p1.9	27505	4680	25159	2192	8.9
p1.10	27003	4440	24592	2177	9.3

To evaluate solution quality and flexibility of the XVRP system, 25 problems were solved with each of five different VRP solvers as well as XVRP. The test problems are fully dense with 100 stop points, have vehicles with a utilization factor of 95 %, and are generated in a square, 1023 miles on each side. The problems were solved for 2, 4 and 8 vehicles which were utilized up to 95% of its maximum capacity and the results for 4 vehicles are shown in Table 10. Examination of Table 10 shows that in all cases XVRP produces the best solution. Percentage improvement over the best of the other algorithms ranged as high as 11.3 percent. Note that the few problems where XVRP did not perform better were because the test was terminated after 1000 trials each time.

Table 10: The values shown are the actual performance of the different algorithms in miles. In all cases but one XVRP-GA produced the best solution. The % of MIN column indicates how XVRPGA performed in comparison with the minimum of the four alternate algorithms. The % of MAX column indicates the how XVRP-GA performed in comparison to the maximum of the four alternate algorithms. The values shown in the table are the total miles traveled using 4 vehicles.

200 Node - 4 Vehicle - 95% Utilization

PRBM	CLRK	FISH1	FISH2	NYWK	XVRP-GA	%MAX	% Min
p1.1	15164	14526	14526	14715	13956	7.966	3.924
p1.2	16047	15137	15137	15041	14625	8.861	2.766
p1.3	15207	14345	14339	14392	14068	7.490	1.890
p1.4	15879	15038	15038	15097	14788	6.871	1.662
p1.5	15612	14648	14648	15057	14564	6.713	0.573
p1.6	15851	14462	14636	14318	13796	12.964	3.646
p1.7	15415	14525	14587	15312	14369	6.786	1.074
p1.8	15544	14849	14849	14977	14484	6.819	2.458
p1.10	15792	15615	15615	15420	13482	12.727	5.210
p1.11	15585	14556	14556	14599	14157	9.163	2.741
p1.12	15647	15226	15226	15289	14762	5.656	3.047
p1.13	15757	15012	15012	14716	14591	7.400	0.849
p1.14	14357	13861	13871	14438	13653	5.437	1.501
p1.15	15507	14877	14877	14456	14201	8.422	1.764
p1.16	15349	15322	15322	15011	14635	4.652	2.505
p1.17	15555	15298	15298	15205	14558	6.410	4.255
p1.18	15586	14611	14611	14373	14190	8.957	1.273
p1.19	15936	14873	14919	15171	14547	8.716	2.192
p1.20	14914	14101	14261	14505	13583	8.925	3.673
p1.21	14856	14512	14512	14462	14098	5.102	2.517
p1.22	16468	15419	15419	15346	14929	9.345	2.717
p1.23	15544	14849	14849	14977	14509	6.659	2.290
p1.24	15280	14512	14512	14184	14015	8.279	1.191
p1.25	15442	14423	14423	14431	14032	9.131	2.711

Further, as shown in Figure 18, each model behaves differently with each data set, and XVRP performs better in each case. This is due to the adaptive capability of XVRP. Table 11 contains descriptive statistics derived from the values shown in Figure 18. The XVRP system has the lowest average miles traveled for all the 25 problem sets, as indicated by the first row in Table 11. The relatively small standard deviation indicates that the solutions obtained through XVRP are consistent and reliable.

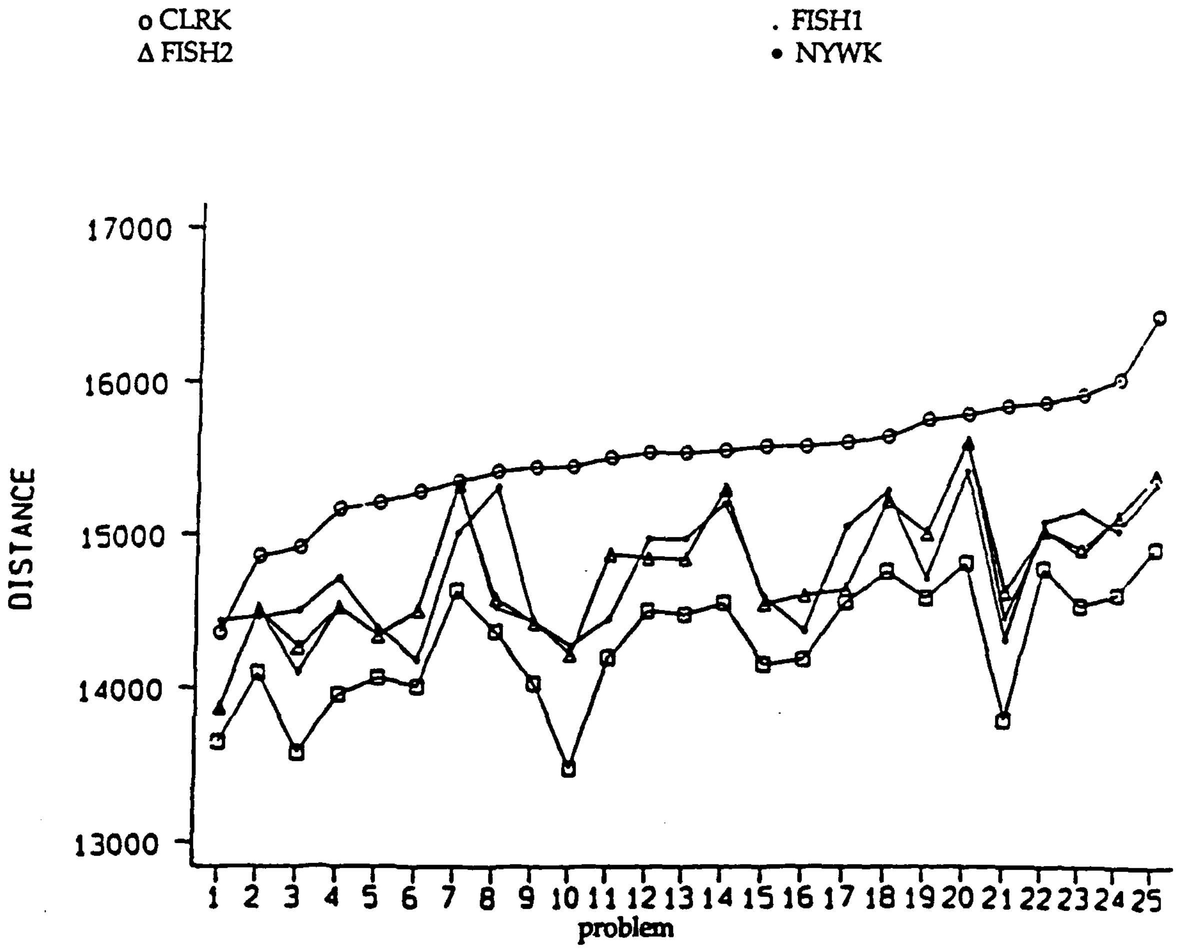
2.4 Conclusions Regarding Genetic Algorithm Routing Techniques

In the vehicle routing problem (VRP), many heuristic algorithms have been developed over the last 25 years. From a purely mathematical standpoint, known methods for finding optimal solutions to most practical routing and scheduling problems require massive computation time even on supercomputers. The studies reported here on adaptively controlling model parameters suggest that there is considerable potential for significantly improving many heuristic algorithms with intelligent shells based on genetic algorithms.

The results demonstrate considerable improvement in the GA search by using multiple evaluation functions. The most original contribution concerns the use of multiple evaluation functions for each population member, computing the fitness of each member to be the maximum returned by any evaluation function. The GA searches many local peaks in parallel in the search space. This establishes a new and promising connection between adaptive search and the relatively mature theory of multi-criteria optimization.

Table 11: The Table contains descriptive statistics of the performance of XVRP-GA.

	CLRK	FISHER1	FISHER2	NYWK	XVRP-GA
Mean	15509	14752	14770	14790	14296
Std. Dev.	420	434	420	391	406
Variance	176903	188602	176690	153396	165063



XVRP-GA Vs other VRP solvers

Figure 18: Each model behaves differently with each data set, and XVRP performs better in each case.

Most investigations into parameter setting within models have been empirical studies that result in static choices for the parameters. The studies reported here on adaptively controlling model parameters suggests that there is considerable potential for significantly improving many heuristic algorithms with intelligent shells based on genetic algorithms. The results are very striking, especially since the five alternative algorithms represent the state-of-the-art from many years of research by Operations Researchers. It is important to note that the method succeeds because it is adaptive. As the generations are simulated, promising geographical patterns survive to subsequent generations, ultimately producing superior parameters for the generalized assignment model. Thus, the overall method finds initial model parameters that are promising, and refines them further even as the overall procedure is running. Finally, since genetic algorithms use solution quality to govern the selection process, the method is ideal for addressing problems in which there are non-standard objectives or several objectives. The uniform superiority of the solutions when compared with the best known mathematical algorithms working alone is most encouraging, and suggest that this type of system can be effectively used as an intelligent shell in any system for which intelligent setting of parameters has potential for significantly improving model performance.

3.0 Set Partitioning Methodology Applied to the Air Force LOGAIR System

Our investigation pursued possibilities of achieving better route solutions in NP hard routing and scheduling problems using genetic algorithms and found an algorithm that was better than the best of the set of state-of-the-art algorithms developed over the last 25 years. In addition to the genetic algorithm approaches previously described, we also developed a unique set partitioning algorithm which derived excellent solutions to NP hard routing and scheduling problems. The general set partitioning problem can be stated as the following zero-one integer program:

> Minimize c_x subject to $E_x = e$ and $x_j = 0$ or 1 ($j = 1,..,n$), where $E = (e_{ij})$ is an m by n matrix whose entries e_{ij} are 0 or 1, $c = (c_j)$ ($j = 1,...,n$) is a cost row with positive components, $x = (x_j)$ ($j = 1,...,n$) is a vector of zero-one variables, and e is an m vector of 1's.

If the $Ex = e$ constraints are replaced by $Ex >= c$ then the above formulation is referred to as a set covering problem. These formulations have been applied to many different problems, including vehicle routing and scheduling. We now show how a set partitioning formulation can be used to aid in the annual design of LOGAIR route structures.

3.1 A Set Partitioning Approach For LOGAIR

In The LOGAIR problem, there are 6 ALCs and 48 other airforce bases. Any routes that are flown between the ALCs are referred to as "trunk" routes and the remaining routes are referred to as "feeder" routes. As in the genetic search, preliminary solutions to the routing problem will be obtained in two phases. There will also be a third phase in which the preliminary solutions are refined and scheduling aspects worked out through user interaction. In the first phase, candidate feeder routes are generated and represented as columns in the E matrix of a set partitioning formulation. As a result of solving this set partitioning problem, a subset of the candidate routes is scheduled such that all bases belong to a feeder route and the total mileage is a minimum. Likewise, in the second phase, candidate trunk routes are generated and a second set partitioning problem is solved to give a trunk route structure with minimal total distance.

The following assumptions are made in the model:

1. Each of the 48 service bases and aerial ports of embarkation belong to only one feeder route. None of the bases are served by multiple routes
2. Each depot or ALC originates zero or more feeder routes.
3. Each aircraft has a home base which it must return to in one flight day.
4. All feeder routes provide service 6 days per week (Mon-Sat).
5. On any given route, pickups and deliveries are mixed.
6. There is no fixed limit on the number of aircraft available.

The input to the system is as follows:

1. A forecast matrix which gives the yearly weight of cargo that must be moved between each pair of bases.
2. The speed and capacity of each type of aircraft.
3. A distance matrix which gives the flight distance between each pair of bases.

3.2 The Feeder Routing Proceedure

In the feeder route phase, a number of candidate feeder routes are generated. This leads to the following set partitioning formulation of the problem:

$$\text{Min } Z = \text{SUM } (c_j)(x_j) \qquad j=1$$

such that

$$\text{SUM}(e_{ij})(x_j) = \text{for } i = 1,\dots,m \qquad j=1$$

$$x_j = 0 \text{ or } 1.$$

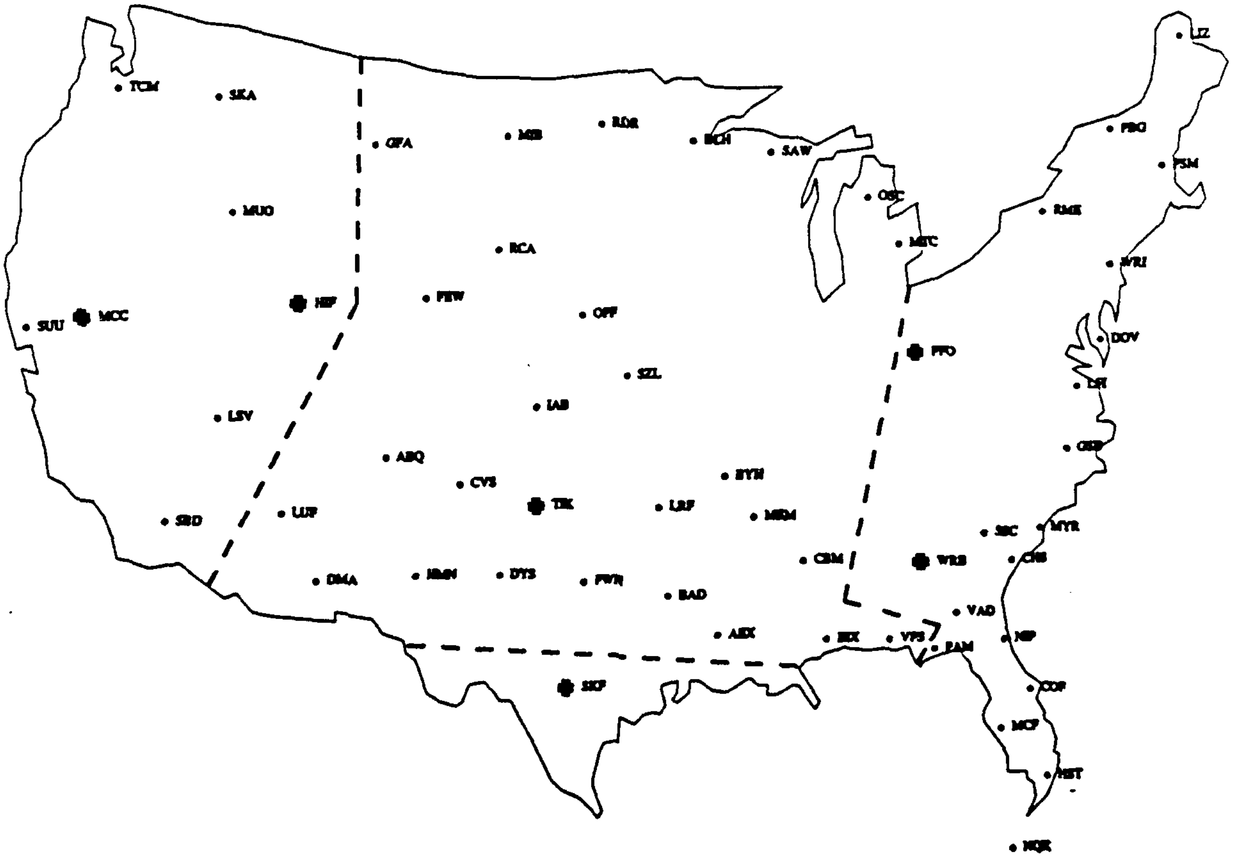


Figure 19. Map Illustrating Constraints Applied to Bases Being Selected for Routes From A Specific Depot. SUU, Is Obviously a Bad Solution For a Route Out Of TIK

In this formulation, n is the number of candidate feeder routes, m is the number of feeder bases, cj is the cost in mileage of candidate route j, and the value of variable xj is 1 or 0 depending on whether candidate route j is selected or not. Each of the candidate routes is represented as a column Ej. The ith element of E_j is one if base i is served by route j and zero otherwise. The set partitioning problem becomes intractable if the number of columns is too large. Therefore, some hueristics are needed for restricting the number of candidate feeder routes. Three restrictions are imposed on the candidate feeder routes. The first restriction is to assign a subset of the feeder bases to each depot or ALC. Any subsequent feeder route which originates from this depot can contain only bases from this set. A feeder base may belong to more than one set. Distance from the depot or ALC is the main criterion for determining set membership. In this way, the number of candidate routes is reduced to a manageable size by eliminating large numbers of obviously poor solutions. An example of this concept is illustrated in Figure 19. Secondly, the number of possible candidate feeder routes is further restricted by setting an upper limit on route distance. Finally, candidate routes which violate user defined limits on aircraft utilization are excluded.

Every time an aircraft flys a route, there is a point along the route in which its load reaches a maximum. If the aircraft is to exclusively deliver cargo, this maximum occurs at the beginning of the route. If the aircraft is to exclusively pickup cargo, this maximum occurs at the end of the route. If, as we assume, the aircraft makes mixed pickups and deliveries, this maximum could occur at any point in the route. Thus, given the forecasted shipping demands, an expected maximum load can be calculated for every prospective route. It is an "expected" value since it is calculated based on daily averages derived from forecasted cargo movements for the year. In order to maximize aircraft utility, the maximum expected load should be close to the capacity of an available aircraft. The LOGAIR personnel decide how close this should be by providing upper and lower tolerance levels on how much expected maximum load may deviate from capacity. These tolerance levels are provided for each type of aircraft. They provide a range of compatibility between routes and aircraft types. If the expected maximum load of a potential route does not fall within the acceptable range of any available aircraft type, the route is rejected. If it falls in the acceptable range of more than one aircraft, the user makes a choice of which aircraft to assign before beginning the trunk route phase. Thus, in order to determine if a candidate route is acceptable, this expected maximum load must be calculated. We introduce some notation that is used in calculating the expected maximum load for a particular route. Let CARGO be a matrix such that CARGO(i,j) denotes the expected daily demand, in pounds, for cargo to be moved from base i to base j. Let B denote the set of all bases, including the ALCs. For a particular feeder route, let R denote the sequence of bases, other than the depot, that belong to this route. Let r belong to R. Let the set P(R,r) denote {s element of R: s precedes r in route R}. The load on the aircraft when it initially takes off from the ALC to any feeder route R can be calculated as follows:

$$L(R) = \underset{\substack{i \text{ element of } B - R \\ j \text{ element of } R}}{\text{SUMMATION}} \text{CARGO}(i,j) + \underset{\substack{i \text{ element of } R \\ j \text{ element of } P(R,i)}}{\text{SUMMATION}} \text{CARGO}(i,j)$$

The first summation term is the total amount of cargo that is to be moved from bases not on the route to bases on the route. All of this cargo is first collected at the depot and then distributed on the next feeder flight. The second summation term is due to the orientation or sequencing of the route and the fact that pickups and deliveries are mixed. If cargo from one base is destined for a base which is visited earlier on the route, it must be routed through the depot for the next flight. The expected load when the aircraft leaves the depot on feeder route R is then L(R). The expected load when the aircraft leaves base i on route R is then given by

$$L(R,i) = L(R) + \text{SUMMATION } (S(k) - D(k)) \qquad k \text{ element of } P(R,i) \text{ U } \{i\}$$

where

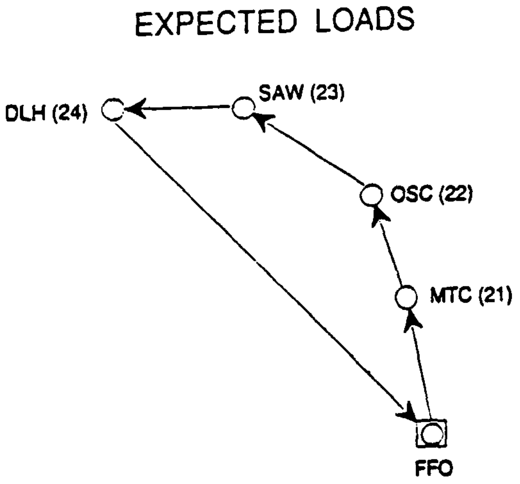
S(i) = SUMMATION CARGO(i,k) (Daily amount of cargo supplied by base i) k element of B

D(i) = SUMMATION CARGO(k,i) (Daily amount of cargo demanded by base i) k element of B

Thus, the aircraft starts out at the depot with expected load L(R). At each base along the way, the load is decremented and incremented by the demand and supply respectively of that particular base. The maximum expected load for a given feeder route R is then given by

MAXL(R) = MAXIMUM (MAXIMUM L(R,i), L(R)) i element of R.

Figure 20 illustrates the procedure for calculating the maximum expected load.



S(i) = supply of cargo at base i
D(i) = demand for cargo at base i

B = index set of all bases
R = feeder route sequence (21,22,23,24)
CARGO(i,j) = requirement for cargo moved from i to j

$$L(R) = \sum_{\substack{i \varepsilon B\text{-}R \\ j \varepsilon R}} CARGO(i,j)$$

$$+ CARGO(22,21)$$
$$+ CARGO(23,21) + CARGO(23,22)$$
$$+ CARGO(24,21) + CARGO(24,22) + CARGO(24,23)$$
$$= 23{,}273 \text{ lbs}$$

L(R,21) = L(R) + S(21) - D(21) = 19,072 lbs
L(R,22) = L(R,21) + S(22) - D(22) = 12,233 lbs
L(R,23) = L(R,22) + S(23) - D(23) = 7,779 lbs
L(R,24) = L(R,23) + S(24) - D(24) = 6,333 lbs

Figure 20. Diagram Illustrates How Expected Load Varies Over a Route

All possible candidate feeder routes that satisfy the 3 restrictions mentioned above are enumerated. Each column contains only information on bases that belong to a feeder route and not the sequencing of bases. Therefore, if there is more than one acceptable route serving exactly the same set of bases, then the route with the least total mileage is chosen for inclusion in the set partitioning model. The mileage cost for each of these routes can be calculated from the distance matrix. The resulting set partitioning problem is solved, yielding a set of feeder routes in which total distance is minimized.

3.3 The Trunk Routing Procedure

The trunk routes are the main arteries of cargo flow in the LOGAIR network. While feeder routes serve to collect and distribute cargo among bases local to a particular ALC, trunk routes serve to transport cargo between ALCs at a national Level. The trunk route phase consists of three steps. The first step consists of obtaining a high quality trunk network. After this step is complete, only the arc capacities of the trunk network are known. No actual routes have been formed. The second step consists of generating candidate routes from the given arcs in the trunk network. The third step consists of solving a set partitioning problem to select the final trunk routes from the set of candidate trunk routes.

The first stop is currently a manual process but could be automated in the future, possibly using a genetic algorithm. Each trunk arc in the network is assigned a capacity which is a linear combination of the capacities of the available aircraft. This is because the arcs have to be broken down into actual flight legs of various trunk routes in the second phase. The arcs forming the feeder routes, whose capacities have previously been determined, together with the candidate trunk arcs, form a network for the MCTP. In the manual process, the user can obtain a high quality trunk network by repeatedly evaluating and modifying a candidate network. The MCTP returns information on arc flows that can be used to find arcs in which an increase or decrease in capacity would be beneficial. If an arc is found to be restricting cargo flow because of inadequate capacity, the capacity may be increased. If the capacity of an arc is not being used efficiently, the capacity may be decreased. The user experiments with various trunk networks until he is satisfied with the amount of cargo moved and the total distance traveled.

In the second step, candidate trunk routes are generated from the set of arcs in the previously determined trunk network. In this process, each trunk arc is actually viewed as a set of flight legs, each with a capacity corresponding to an available aircraft type. The sum of the capacities of the flight legs is crucial to the capacity of the arc. The trunk routes are formed by constructing feasible sequences of flight legs. Each route generated will produce a column in the set partitioning problem that is to be solved in the third step. Again, to limit the number of columns generated, three restrictions are enforced. The first restriction is that routes which contain more than one flight leg have to begin and end at the same ALC. In routes consisting of just one flight leg, it is assumed that the aircraft carries cargo in the direction of the flight leg and immediately flys back empty. The second restriction is that all the flight legs which make up the route must have the same capacity. Finally, a limit is placed on route distance. All routes that satisfy these restrictions are incorporated into the set partitioning formulation. The cost associated with each route is the amount of time by which the flight time deviates from a target flight time set for trunk routes. Since the speed, determined by aircraft type, and distance are known for each route, the time required to fly the route can be calculated.

The final step is to solve the set partitioning problem to obtain the final set of trunk routes. The set partitioning formulation is as follows:

$$\text{Min } Z = \text{SUM } (c_j)(x_j) \qquad j=1$$

such that

$$\text{SUM}(e_{ij})(x_j) = 1 \text{ for } i = 1,\ldots,m \qquad j=1$$

$$x_j = 0 \text{ or } 1.$$

In this formulation, n is the number of candidate trunk routes, m is the number of flight legs, cj is the cost of candidate route j as described above, and the value of variable xj is 1 or 0 depending on whether candidate route j is selected or not. Each of the candidate routes is represented as a column Ej. The ith element of E_j is 1 if flight leg i is in route j and 0 otherwise. The solution to this problem is a set of trunk routes such that each flight leg belongs to exactly one trunk route and the length of each route is close to the desired target value.

4.0 Graphical User Interface for LOGAIR Routing and Scheduling Package Menu Functions and Descriptions

A graphical user interface has been designed to provide an environment which enhances the users ability to construct good routing solutions. It is a fully menu driven system. All menu options are selected by using a mouse to point to the desired selection. It has many features which simplify the activities involved in the route development process. We now describe these features in a format which reflects the hierarchical structure of the menu system. The menu titles are listed along with a functional description of each. There are seven main menu selections, each of which is broken down into submenus as follows:

MAIN MENU

1) BASE
2) STRUCTURE
3) MODIFY
4) VIEW
5) OPTION
6) PROCEDURE
7) QUIT

SUBMENUS

1) LOGAIR Base Graphics

    1.1 Add a New Base
    1.2 Remove an Old Base
    1.3 Modify Tonnage Matrix
    1.4 Modify Aircraft Characteristics

2) Route Structure Manipulation

    2.1 View Load Solutions
    2.2 Save Route Solution
    2.3 Delete Unwanted Route Solutions
    2.4 Enlarge Display
    2.5 Report Generation
    2.6 Print Reports

3) Modify Route Structure

    3.1 Load Existing Route
    3.2 New Route Generation
    3.3 Link New Route With Existing Structure
    3.4 Remove Existing Route Segment

4) View Route Route Segment

    4.1 Route Color Assignment
    4.2 Aircraft Icon Assignment
    4.3 Ratio of Planned Leg Cargo Loading of Aircraft as a Function of Total Carrying Capacity
    4.4 Animation of Route Structure Cargo Handling Capabilities

5) User Customization Features

5.1 Background Color Selection
5.2 USA Map Color Selection
5.3 Display Base Name Switch
5.4 Select Trunk Route Color
5.5 Trunk Route Line Width Selection
5.6 Feeder Route Color Selection
5.7 Feeder Route Line Width Selection

6) Selection of Active Route Optimization Method

6.1 Genetic Algorithms
6.2 User Selection of Set Partitioning Algorithm
- 6.2.1 User Specified Route Constraints
- 6.2.2 Run Set Partitioning Problem
- 6.2.3 Final Feeder Route Structure
- 6.2.4 Trunk Route Intermediate Solution Display
- 6.2.5 Detailed Feeder Route Load and Stop Information
- 6.2.6 Cargo Movement Between ALCs
- 6.2.7 Aircraft Selection For Route Segment

The next sections provide brief descriptions of the different functions of the graphical user interaface.

4.1 LOGAIR Base Graphics

The selections under this main option provide the capability to add, delete and update all the information which is relavent to the LOGAIR routing problem. (Note: Not yet completed at this time).

4.1.1 Add a New Base

This selection permits the user to add a new base to the system. This is accomplished by moving the mouse to the desired location and adding the base. The system then prompts the user for the required information for the new base.

4.1.2 Remove an Old Base

This selection permits the user to remove an existing base from the system. The desired base is located with the mouse and subsequently deleted.

4.1.3 Modify Tonnage Matrix

This selection permits the user to modify the tonnage or demand matrix to reflect changes in shipping requirements.

4.1.4 Modify Aircraft Characteristics

This selection permits the user to maintain information on available aircraft types and their characteristics. Aircraft types may be added or deleted as well as updating information on existing aircraft types. Aircraft type attributes include capacity, speed, fuel-burn rate, etc.

4.2 Route Structure Manipulation

The selections under this main option provide the capability to store, retrieve and compare various routing solutions.

4.2.1 View Load Solutions

This selection permits the user to view up to 3 solution structures simultaneously. This feature enables the user to compare and contrast alternative solutions in an efficient manner. The screen is divided into 4 equal portions, 3 for the actual graphic displays of solutions and 1 for the list of solutions which are available for viewing.

4.2.2 Save Route Solution

This selection permits the user to save various routing solutions. Any solution generated on the system may be saved, irreguardless of the method in which it was obtained. The method may be an algorithm supported by the system, or the manual construction or modification of a routing solution. The routing solution is displayed before being stored in the system.

4.2.3 Delete Unwanted Route Solutions

This selection permits the user to delete any existing routing solution.

4.2.4 Enlarge Display

This selection permits the user to enlarge the display of any solution that is currently displayed. The available scaling factors are 125, 150, 175 or 200 percent.

4.2.5 Report Generation

This selection provides the user with detailed information for each route of a particular solution. This information includes the sequence of bases on each route, the expected load at each base, the remaining available space, total mileage and times of arrival and departure.

4.2.6 Print Reports

This selection permits the user to obtain printed reports and maps for a particular routing solution. (Note: Not yet completed at this time).

4.3 Modify Route Structure

The selections under this main option provide the capability to modify existing solutions or create new ones.

4.3.1 Load Existing Route

This selection permits the user to load an existing solution that he or she wishes to modify. The user is presented with a list of all stored solutions from which to choose.

4.3.2 New Route Generation

This selection permits the user to create a route structure solution from scratch. The user is prompted for a name under which the constructed solution will be stored.

4.3.3 Link New Route With Existing Structure

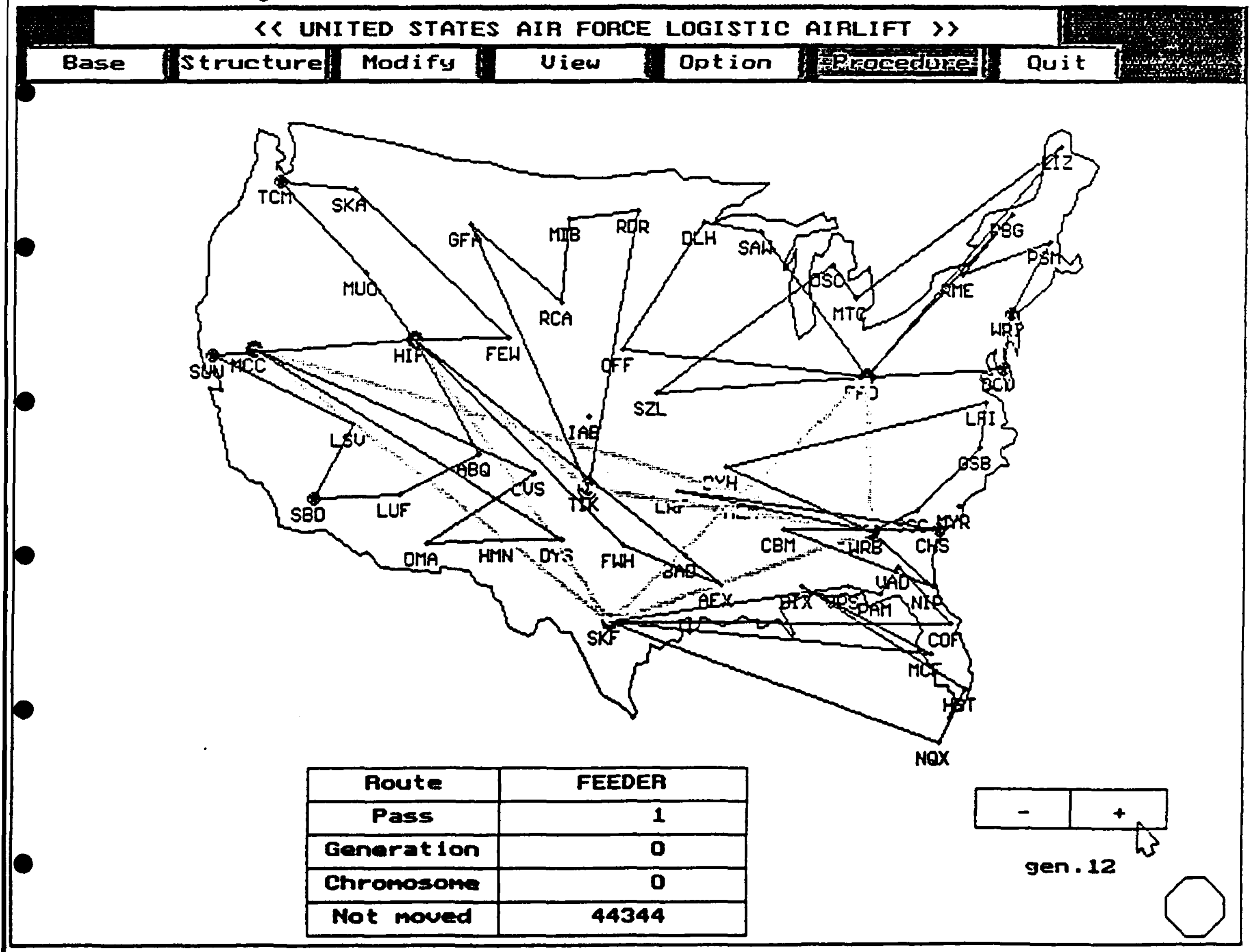
This selection permits the user to add links or flight legs to a route by using the mouse to point to the starting base and ending base for each new link.

4.3.4 Remove Existing Route Segment

This selection permits the user to remove links or flight legs from a route. The mouse is used to point to the link that is to be removed or disconnected.

4.4 View Route Segment

The selections under this main option provide the capability to graphically highlight various aspects of a particular routing solution. This is accomplished by using a color coding scheme to illustrate the particular aspect of the routing solution that is currently of interest. An example of this concept is shown in Figure 21.



Route	FEEDER
Pass	1
Generation	0
Chromosome	0
Not moved	44344

Figure 21. Sample Screen Of Route Modification Section From Graphical User Interface

4.4.1 Route Color Assignment

This selection enables the user to easily distinguish between the individual routes flown by displaying each route in a different color.

4.4.2 Aircraft Icon Assignment

This selection presents the user with a display which gives a good visualization of the types of aircraft that are being used for each individual route. Each route is color coded according the aircraft type which is used for that particular route.

4.4.3 Ratio of Planned Leg Cargo Loading of Aircraft as a Function of Total Carrying Capacity

This selection presents the user with a display in which he or she can visualize the changing cargo load on the aircraft as it visits the bases along its route. This is accomplished by color coding the individual flight legs of the routes in terms of the amount of cargo that is on board the aircraft for each flight leg. Each different color represents a different load range. The different ranges occur in increments of ten thousand pounds of load per day.

4.4.4 Animation of Route Structure Cargo Handling Capabilities

This selection presents the user with a simulation of aircraft movement. The user can step through time in 30 minute intervals and see the location of each aircraft at each interval. This simulation covers a 24 hour period.

4.5 User Customization Features

The selections under this main option provide the user with the capability of customizing various display attributes to suit his or her taste.

4.5.1 Background Color Selection

This selection permits the user to select the color in which the background is displayed. One of 16 different colors may be selected.

4.5.2 USA Map Color Selection

This selection permits the user to select the color in which the USA map is displayed. One of 16 different colors may be selected.

4.5.3 Display Base Name Switch

This selection permits the user to specify if the bases are to be labled on the map display.

4.5.4 Select Trunk Route Color

This selection permits the user to select the color in which the trunk routes are to be displayed. One of 16 different colors may be selected.

4.5.5 Trunk Route Line Width Selection

This selection permits the user to select the line width in which the trunk routes are to be displayed. There are 2 choices for line width.

4.5.6 Feeder Route Color Selection

This selection permits the us·r to select the color in which the feeder routes are to be displayed. One of 16 different colors may be selected.

4.5.7 Feeder Route Line Width Selection

This selection permits the user to select the line width in which the feeder routes are to be displayed. There are 2 choices for line width.

4.6 Selection of Active Route Optimization Method

The selections under this main option provide the user with algorithm support for creating routing solutions. The types of algorithms used are genetic algorithms and set partitioning algorithms.

4.6.1 Genetic Algorithms

This selection permits the user to access the genetic algorithms that were developed for the LOGAIR route design problem. (Note: This part is not yet user interactive).

4.6.2 User Selection of Set Partitioning Algorithm

This selection permits the user to access the set partitioning algorithms that were developed for the LOGAIR route design problem.

4.6.2.1 User Specified Route Constraints

This selection permits the user to specify various constraints that are to be imposed on the route structures that are to be generated. These constraints include the available aircraft types, maximum mileage for a single route, and restrictions on aircraft utilization.

4.6.2.2 Run Set Partitioning Problem

This selection permits the user to view intermediate results at various stages of the algorithm. At each iteration, it shows a new set of feeder routes being added to the final solution.

4.6.2.3 Final Feeder Route Structure

This selection shows the final solution of feeder routes generated by the Set Partitioning algorithm.

4.6.2.4 Trunk Route Intermediate Solution Display

This selection permits the user to view intermediate results at various stages of the algorithm. At each iteration, an improved trunk route structure is displayed.

4.6.2.5 Detailed Feeder Route Load and Stop Information

This selection gives the user a detailed report on the feeder routes of the final solution. The information on each route includes the route sequence as well as the expected load at each base along the route.

4.6.2.6 Cargo Movement Between ALCs

This selection provides the user with information as to how much cargo must be moved between each pair of ALCs. These required cargo movements are dependent on the previously determined feeder route structure. This information may then be used to develop
the structure of the trunk routes.

4.6.2.7 Aircraft Selection for Route Segment

This selection provides the user with a list of aircraft that are compatible with each feeder route.

5.0 Future Research Directions

During Phase I, two primary methodologies specifically for LOGAIR route design were developed, programmed and evaluated. The first is based on genetic search, and is inspired by the basic research described above. The second utilizes a set partitioning method, an approach well-known to be effective in highly constrained routing and scheduling problems. In addition, a graphical user interface (GUI) specific to LOGAIR was prototyped and demonstrated to LOGAIR personnel.

During Phase II, Netrologic, with the participation of consultant Dr. Kendall E. Nygard at NDSU, will extend both types of work explored and accomplished in Phase I. In particular, the multi-paradigm approach will be extended to address three very difficult and important problems in vehicle routing: The multi-depot problem, the two-sided time window problem, and the dynamically changing parameters routing problem. These complexities, despite many man-centuries of effort, are not fully resolved to date. In addition, the Phase II effort will apply the methods to the LOGAIR problem, and complete the software system for use by LOGAIR personnel in their route design process. We will deliver reports, journal articles and presentations on all of the work and deliver User Manuals and Programmer Reference Manuals for the completed LOGAIR software system. We have every expectation that the Phase II effort will be highly significant in advancing the state-of-art in routing and scheduling, and result in a software product that will provide a useful service to LOGAIR, and to extend the basic understanding of optimal routing and scheduling.

The proposed Phase II project will be a significant advance in basic research concerning routing and scheduling, and also be responsive to a clear need of the Air Force through LOGAIR route design. There is potential for the results of this research to apply to many other military and commercial transportation systems, including the operations of the Military Airlift Command.

References

Baker, J.E., *Adaptive Selection Methods for Genetic Algorithms*, Proceedings of the First International Conf. on Genetic Algorithms and their Applications, July 24-26, Pittsburgh, 101-111, 1985.

Bodin, L., Golden, B.R., Assad, A., and Ball, M., *Routing and Scheduling of Vehicles and Crews*, Computers and Operations Research, 10, 62-212, 1983.

Clarke, G. and Wright, J., *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points*, Operations Research, 12 (4), 568-581, 1964.

Fisher, M., and Jaikumar, J., *A Generalized Assignment Heuristic for Vehicle Routing*, Networks, 11, 109-124, 1981.

Goldberg, D., *Genetic Algorithms with Sharing for Multimodal Function Optimization*, Proceedings of the Second International Conf. on Genetic Algorithms, July 28-31, MIT, 41-49, 1987.

Coldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989

Goldberg, D., *Sizing Populations for Serial and Parallel Genetic Algorithms,* Proceedings of the Third International Conf. on Genetic Algorithms, June 4-7, George Mason University, 70-79, 1989.

Grefenstette, J., *Parallel Adaptive Algorithms for Function Optimization (Preliminary Report),* Technical Report CS-81-19, Computer Science Department, Vanderbilt University, 1981.

Grefenstette, J., *GENESIS: A System For Using Genetic Search Procedures,* Proceedings of the 1984 Conf. on Intelligent Systems and Machines, 161-165, 1984.

Hayong, Z., *Classifier System With Long-term Memory in Machine Learning,* Proceedings of the First International Conf. on Genetic Algorithms and Their Applications, July 24-26, Pittsburgh, 178-182, 1985.

Holland, J., Adaptation in Natural and Artificial Systems, Ann Arbor, MI, University of Michigan Press, 1975.

Jog, P., Suh, J.Y., and Gucht, D.V. *The Effects of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem,* Proceedings of the Third International Conf. on Genetic Algorithms, June 4-7, George Mason University, 110-115,1989.

Kadaba, N. (a), *XROUTE: A Knowledge-Based Routing System using Neural Networks and Genetic Algorithms,* Ph.D. Dissertation, Dept. of Computer Science and Operations Research, North Dakota State University, Fargo, 1990.

Kadaba, N. (b), *Using Genetic Algorithms to train Back-propagation Neural Networks.* NDSU Tech. Report NDSU-CS-90-19-44, 1990.

Kadaba, N., Nygard, K.E. (a), *Using Genetic Algorithms as a Post-Processor for Improving Vehicle Routing Solutions,* 5th Rocky Mountain Conference on Artificial Intelligence (RMCAI-90), Las Cruces, New Mexico, June 28-30, 1990.

Kadaba, N., Nygard, K.E. (b), *Improving the Performance of Genetic Algorithms in Automated Discovery of Parameters, to appear in the Proceedings of the International Conference on Machine Learning, Austin, June, 1990.*

Kadaba, N., Nygard, K.E., and Juell, P.L., *Integration of Knowledge-Based System and Adaptive Learning Techniques for Routing and Scheduling Applications,* forthcoming in Expert Systems with Applications: An International Journal, 1990.

Kadaba, N., Perrizo, W., Ram, P., *Performance Analysis of Distributed Systems for Parallelizing Genetic Algorithms,* NDSU Technical Report NDSU-CS-TR-904, 1990. Submitted for publication.

Nelson, M.D., *Implementation Techniques for the Vehicle Routing Problem,* Master's thesis, North Dakota State University, Fargo, North Dakota, May, 1983.

Nelson, M.D., K.E. Nygard, J.H. Griffin and W.E. Shreve, Implementation Techniques for the Vehicle Routing Problem, 1983.

Nygard, K.E. and Kadaba, N. *Modular Neural Networks and Distributed Adaptive Search for Travelling Salesman Algorithms,* Proceedings of SPIE Technical Symposium on Optical Engineering and Photonics in Aerospace Sensing, Orlando, Florida, April 16-20, 1990.

Nygard, K.E., Greenberg, P., Bolkan, W. and Swenson, E., *Generalized Assignment Methods for the Deadline Vehicle Routing Problem,* Vehicle Routing: Methods and Studies, B. Golden and A. Assad, Eds., NorthHolland, 1988.

Nygard, K.E., Juell, P. and Kadaba, N., *Artificial Intelligence in Routing and Scheduling Applications,* Proceedings of the 4th Aerospace Applications of Artificial Intelligence Conference, Dayton, October, 1989.

Nygard, K. E. and R. Walker, *A Vehicle Routing Algorithm Based on Spacefilling Curves, Technical Report, Department of Computer Science and Operations Research, North Dakota State University, Fargo, N.D., 1988.*

Rumelhart, D. E., and McClelland, J., Parallel Distributed Processing, Explorations in the Microstructure of Cognition, MIT Press, Cambridge, 1986.

Schaffer, D., Caruana, R.A., Eshelman, L.J., and Das, R. *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization,* Proceedings of the Third International Conf. on Genetic Algorithms, June 4-7, George Mason University, 51-60, 1989.